komputer

7-12

\1990\

popularny miesięcznik informatyczny: \#(50)90\ cena 5000 zł.

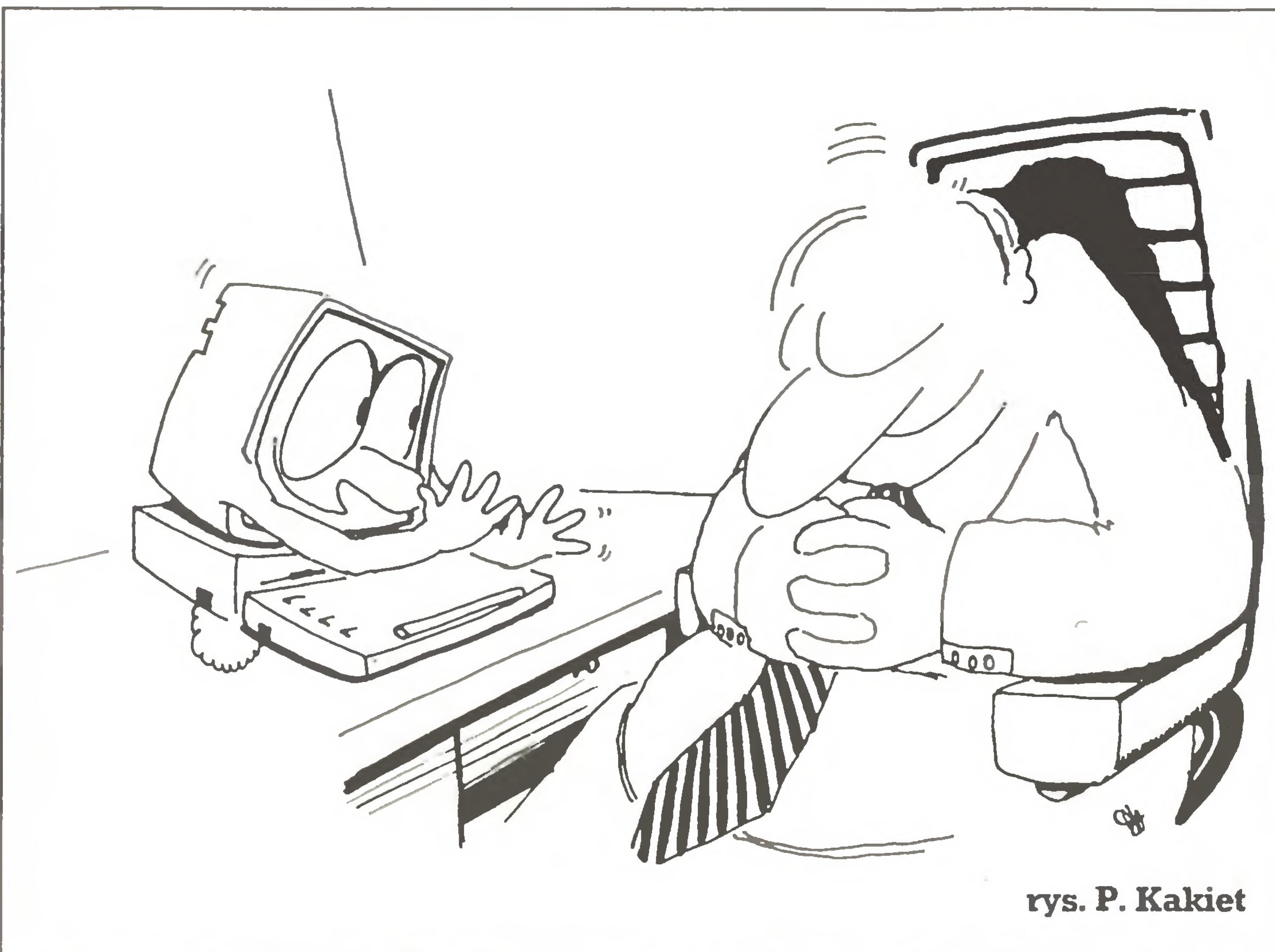


KOMPUTER NA EMERYTURZE

rys. Piotr Kakiet

Nad pięćdziesięcioma numerami miesięcznika „Komputer” pracowali:

Włodzimierz Banaszak
Andrzej Bączyński
Zbigniew Blewoński
Elżbieta Bobrowska
Maciej Borkowski
Rafał Brzeski
Marek Car
Grzegorz Czapkiewicz
Mariusz Dec
Grzegorz Eider
Leszek Gołębiowski
Jacek Jedynak
Tomasz Jedynak
Andrzej Kadlof
Piotr Kakiet
Jarosław Kania
Agnieszka Kasprzycka
Zbigniew Kasprzycki
Zbigniew Kondraciuk
Stanisław Królak
Krzysztof Krupa
Krzysztof Kuryłowicz
Jacek A. Likowski
Małgorzata Luzińska
Władysław Majewski
Beata Maruszewska
Krzysztof Matey
Marek Matuszczak
Tomasz Mazur
Romualda Miarecka
Wiesław Migut
Jarosław Młodzki
Marek Młynarski
Adam Nowicki
Wojciech Olejniczak
Maria Omiecińska
Mariusz Pietruszka
Andrzej J. Piotrowski
Sergiusz Piotrowski
Andrzej Popławski
Jerzy Pusiak
Izabela Radzikowska
Juliusz Rawicz
Leszek Rudak
Zenon Rudak
Michał Setlak
Magdalena Stachorzyńska
Jan Stożek
Stefan Szczypka
Grzegorz Szewczyk
Anna Szubka
Jakub Tatarkiewicz
Darosław J. Toruń
Piotr Tymochowicz
Irena Urbaniak
Roland Wacławek
Tadeusz Wilczek
Wojciech Wojtanowski
Andrzej Załuski
Tomasz Zieliński


rys. P. Kakiet

7-12 (50)

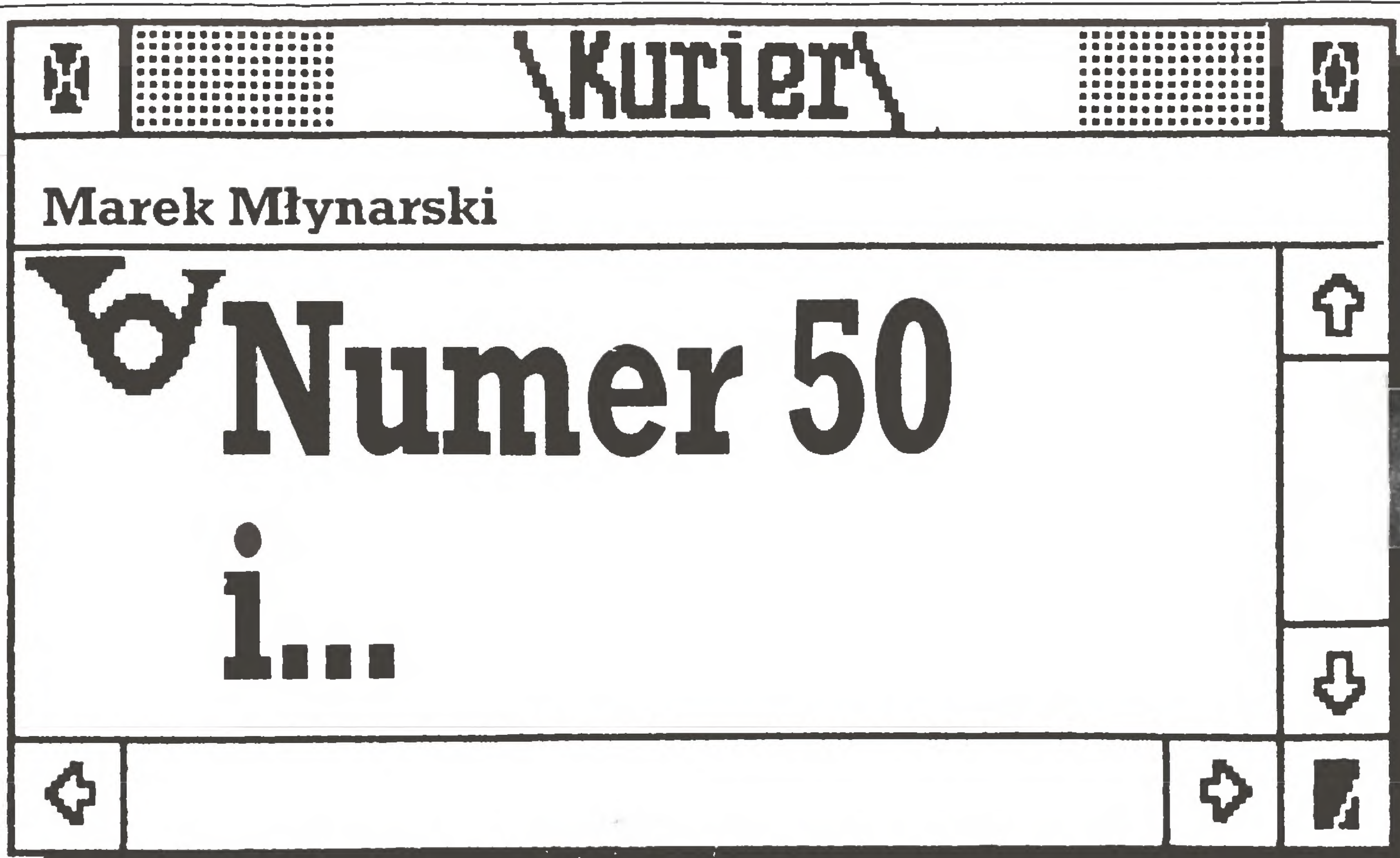
"Komputer" Popularny Miesięcznik Informatyczny - pismo miłośników i użytkowników mikrokomputerów redagują:

Marek Młynarski (red. nacz. tel. w. 330)
Marek Car (z-ca red. nacz. tel. w. 330)
Krzysztof Matey (sekr. red. tel. w. 330)
Kierownicy działów:
Małgorzata Luzińska (dział techn. tel. w. 310)
Zenon Rudak (sprzęt tel. w. 310),
Stefan Szczypka (dział graf. tel. w. 329),
oraz
Halina Bulińska (administracja), Halina Car (wyd. rosyjskie), Grzegorz Czapkiewicz, Maria Omiecińska (korekta), Andrzej Popławski (wyd. rosyjskie), Magdalena Stachorzyńska (operatorka).
Współpracują:
Zbigniew Blewoński, Tadeusz Jedynak, Andrzej Kadlof, Tomasz Mazur, Mariusz Pietruszka, Leszek Rudak, Michał Setlak (Fido), Jakub Tatarkiewicz.

Adres redakcji:
ul. Koszykowa 6a
00-564 Warszawa,
Telefony:
21-19-85 (godz. 10 – 16)
lub centrala 28-22-01 wew. 243 i 328
telex 812405 ruch pl

Sieć FIDO
21 19 85 w godz. 16^{00} – 10^{00}
soboty i niedziele czynne całą dobę.
Wydawca:
Warszawskie Wydawnictwo Prasowe RSW "Prasa-Książka-Ruch",
dyr. Maciej Hoffman,
Al. Jerozolimskie 125/127,
02-017 Warszawa
tel. centrali: 28-52-31
Skład i druk:
Prasowe Zakłady Graficzne,
90-950 Łódź,
ul. Armii Czerwonej 28, tel. centrali 74-74-20
Cena: 5000,- zł. Zam. 2028/90, A-48

Nr indeksu 36-345 ISSN 0860-2514



NASZ – czyli Czytelników i redakcji – miesięcznik „Komputer" od początku uważaliśmy za platformę wymiany informacji według równoprawnych zasad Czytelnicy – Redakcja. Dziś, kiedy z satysfakcją oglądamy wydane numery, można bez fałszywej skromności stwierdzić, że nasz główny cel został osiągnięty. „Komputer" stanowi taką platformę. I chociaż wszyscy wielokrotnie popełnialiśmy błędy – i my – jako redakcja, i Wy – Czytelnicy – wszyscy jesteśmy mądrzejsi niż na początku.

A rozpoczęło się to we wrześniu 1985 roku, kiedy zaczęliśmy wydawać „Bajtka", wówczas jeszcze na gazetowym papierze. Właściwe narodziny przypadły na kwiecień 1986 roku, gdy ruszył nasz „Komputer" już jako kolorowy miesięcznik. Pismo powstawało dzięki uporowi kilkunastu ludzi i pracy kilkudziesięciu innych, których nazwiska wymienione są obok. Początkowy kształt merytoryczny zawdzięcza Władkowi Majewskiemu, nota bene również pomysłodawcy tytułu „Bajtek". Kilkadziesiąt numerów później Władka pochłonął świat wielkiego biznesu. Dziś występuje on w podwójnej roli dziennikarza – biznesmena i zastępcy redaktora naczelnego „COMPUTERWORLDa".

Kilka słów o zastępcach redaktora naczelnego. Przez krótki czas był nim Andrzej Piotrowski – twórca i późniejszy naczelny ś.p. „Mikroklanu", a obecnie szef fundacji chcącej wyciągnąć z bagna krajową sieć telefoniczną. Następny – Grzegorz Eidr – przeszedł kolejne szczeble redakcyjnej kariery, praktycznie jednoosobowo stworzył Katalog Firm Komputerowych, wreszcie utworzył własne pismo – „PC Kurier". Last, but not least – Marek Car – jest twórcą rosyjskiej edycji „Komputera". Jemu też, w dużej mierze, zawdzięcza swą obecną postać polski „Komputer".

Kształt naszego pisma to nie tylko jego zawartość, ale i jego szata graficzna. Od początku czuwa nad nią Stefan Szczypka, najwybitniejszy dziś specjalista w dziedzinie komputerowego kreowania czasopism. Niewdzięczna harówka związana z opracowaniem technicznym „Komputera" i jego reklam jest udziałem Małgosi Luzińskiej. Śmiesznymi (choć nie zawsze zrozumiałymi) rysunkami ozdabiał nasze łamy i na nich też stawiał swe pierwsze kroki Piotr Kakiet.

Sekretarz redakcji – praca trudna i odpowiedzialna, sprowadzająca się m.in. do tłumaczenia tekstów z komputeropolskiego na polski i pilnowaniu wszelkich terminów. Elżbieta Bobrowska jest dziś reporterem w telewizyjnym „Kurierze Warszawskim", G. Eider – patrz wyżej, Stanisław Królak pisze w tygodniku „Gazeta i Nowoczesność", w dzisiejszym „KOMPUTERZE" sekretarzuje Krzysztof Matey, pełniący zarazem obowiązki zastępcy red. nacz. dwutygodnika „COMPUTERWORLD".

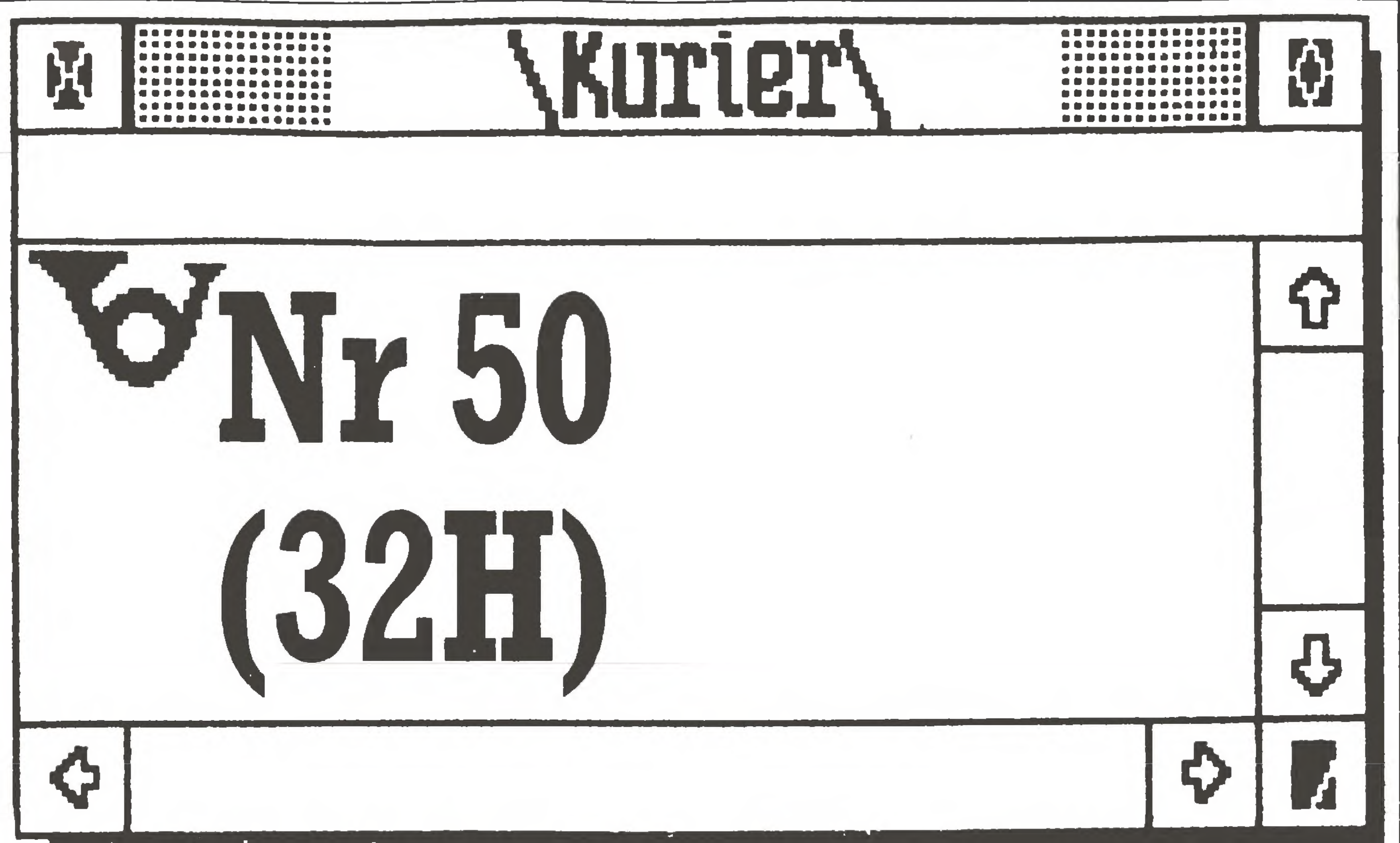
Kierownicy działów – piszący, zbierający i segregujący materiały do poszczególnych działów pisma – co z nimi dzisiaj? Zenon Rudak (obecnie także red. naczelny „COMPUTERWORLD") prowadzi swoją działkę sprzętową. Grzegorz Czapkiewicz jest z nami nadal, ale na pozycji „wolnego strzelca". Tomasz Zieliński i Mariusz Dec – pracują w innym piśmie i w innych interesach.

Obsługą komputera w „KOMPUTERZE" zajmowała się Magdalena Stachorzyńska (dziś w Studio Q Tomka Kuczborskiego), a z bezlikiem czynności kasowych, finansowych i administracyjnych wciąż jeszcze walczy Halina Bulińska.

Dlaczego o tym piszę? Otóż „Komputer" zmienia wydawcę – zamyka się pewien okres w życiu pisma, inaczej zostanie zorganizowana praca. Te dotychczasowe 50 numerów jest dziełem wielu osób, ale ja, jako redaktor naczelny, uważam za konieczne podkreślenie zasług tych, o których napisałem powyżej.

Wszystkim, absolutnie wszystkim, którzy swoją pracą oraz pomysłami przyczynili się do wydania 50 numerów „Komputera" gorąco i szczerze DZIĘKUJĘ.





Gdy pismo ukazuje się rzadziej, jubileusze gęstnieją. Częściej piszemy o sobie, miast o problemach czytelników. Obecna okazja jest przypadkowa, jak wybrany do jej zapisania układ dziesiętny (równie dla nas swojski szesnastkowy ukazuje powszedniość tej liczby...).

Ale... 50 numer "Komputera" jest zarazem ostatnim publikowanym przez Warszawskie Wydawnictwo Prasowe, agendę likwidowanej RSW "Prasa-Książka-Ruch". Następny podpisze nowy wydawca. Gdy **(7 sierpnia)** piszę te słowa, nie wiemy kto nim będzie i jaką firmą będzie się sygnował.

Pewne jest, że będzie co podpisywać. 5-letni dorobek oceniono – dosłownie – wysoko: mimo iż pismo w roku obecnym nie przyniesie zysków, są w Polsce i zagranicą chętni, gotowi sowicie zapłacić za prawo jego kontynuacji. "Komputer" jest więc potrzebny i wart inwestycji.

Oczywiście "Komputer" będzie – mamy nadzieję, że raczej ewolucyjnie niż rewolucyjnie – zmieniał swój kształt. Co z jego dorobku przetrwa, co warte jest przypomnienia?

W "Komputerze" wielokrotnie wyprzedzaliśmy społeczną świadomość znaczenia problemów takich, jak jednolitość kodowania polskich liter. To na naszych łamach ukazał się raport, który z czasem doprowadzi do ustanowienia narodowego standardu. Na czas ostrzegaliśmy przed wirusami, obszernie informując o objawach ich działania i sposobach walki z zagrożeniem. Tu uparcie lansowano ideę porozumiewania się użytkowników przez modemy i system amatorskiej poczty elektronicznej. Próby analizy rynku komputerowego zaowocowały Katalogiem Firm Komputerowych. Lansując cywilizowane formy dystrybucji oprogramowania rozpowszechniamy programy typu freeware i shareware. Naszym udziałem była też licencjonowana produkcja i sprzedaż (ochrona praw autorskich) polskich i zagranicznych programów dla ZX Spectrum. Mamy też twórczy dorobek w komputerowej fazie dziejów kultury słowa drukowanego. Jesteśmy współwydawcą największego pisma komputerowego w ZSRR.

To tylko kilka przykładów z przeszłości. Czy możemy dziś przewidzieć, o czym będziemy pisać przez 50 następnych numerów? Zaryzykuję typowanie:

– wzrośnie znaczenie kwestii społeczno-prawnych zwiazanych nie tylko z interesami informatyków (prawna ochrona oprogramowania), ale i zwykłych użytkowników (ochrona prywatności, bezpieczeństwo danych);

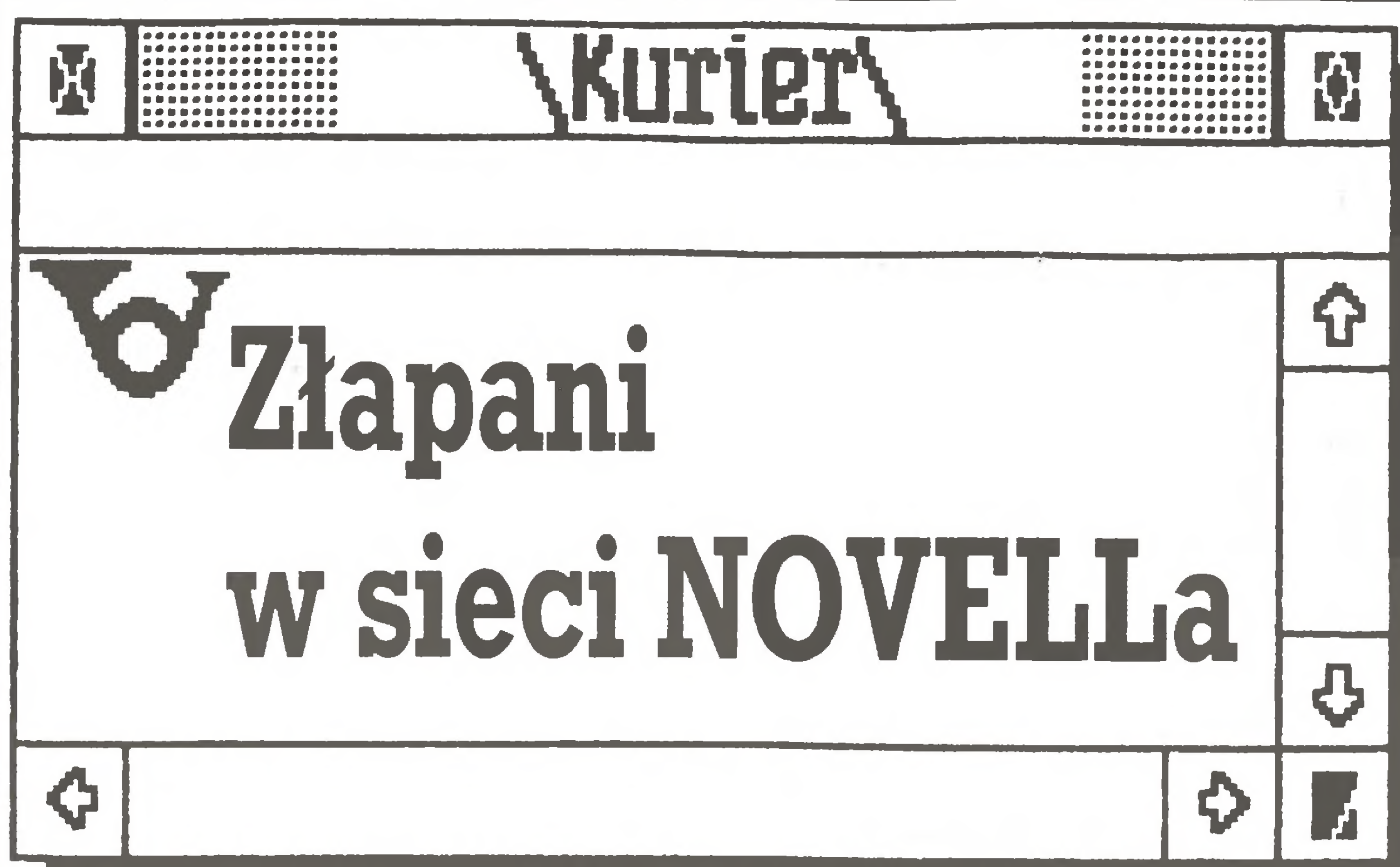
– od strony technicznej ciężar rozwoju przesunie się na techniczną doskonałość kontaktu z użytkownikiem: kolor, animacja, stereo, symulacja i rozumienie mowy;

– programy będą musiały nauczyć się pracy równoległej na wszystkich poziomach.

Czy sprawdzi się? Zobaczymy...

Zespół Redakcyjny ręką **Władysława Majewskiego.**





Z Andrew Żółtowskim, dyrektorem NOVELL UK Ltd. rozmawia (nie tylko o sieciach NetWare) Marek Car

– Chociaż, jak mi się wydaje, doświadczonych czytelników "Komputera" nie trzeba zapoznawać z producentami najpopularniejszego na świecie sieciowego systemu operacyjnego Advanced NetWare 286, to mimo wszystko na początku naszej rozmowy proszę o kilka słów o firmie NOVELL.

– Od 1985 r. nasza firma specjalizuje się w sieciach lokalnych. Jako pierwsi zdecydowaliśmy się zastosować w mikrokomputerach koncepcję pracy file servera (węzeł sieci lokalnej, wykonujący określone funkcje na potrzeby innych węzłów). Przedtem, na tym poziomie techniki komputerowej, użytkownicy jednego twardego dysku zmuszeni byli dzielić się nim w dosłownym znaczeniu tego słowa – zarówno jego objętością jak i czasem dostępu. Opracowany przez nas sieciowy system operacyjny zapewnia dostęp kilku użytkownikom jednocześnie do programów lub do pakietów danych, przeznaczonych do ogólnego użytkowania i przechowywanych na twardym dysku file servera. Za nami poszli inni, ale NOVELL zdążył już zdobyć 90% światowego rynku sieciowych systemów operacyjnych. Według danych z początku 1990 r. na świecie pracuje pół miliona naszych sieci, obsługujących 5 milionów użytkowników. Produkowane przez firmę NOVELL towary są sprzedawane na całym świecie, w tym także w krajach Europy Wschodniej.

– A w Związku Radzieckim?

– W 1989 r. zadecydowaliśmy, że jest to jeden z najbardziej perspektywicznych rynków zbytu dla naszych produktów. Użytkownikom radzieckim proponujemy wszystkie wersje naszych systemów operacyjnych, w tym również najnowsze: NetWare 386, Advanced NetWare v 2.15 i jego uproszczoną wersję dla początkujących ELS (Entry Level Solution) NetWare Level II v. 2.15.

– Zazwyczaj NOVELL nie sprzedaje swoich wyrobów bezpośrednio użytkownikom. Tymi sprawami zajmują się dealerzy bądź autoryzowane przez Was firmy, reprezentujące interesy NOVELLa.

– Dostawcami naszych systemów w Polsce jest kilka firm – ściśle przestrzegamy zasady, by nikt nie miał monopolu i nie otrzymywał wyłączności na dystrybucję naszych produktów na danym rynku.

– Czy władze amerykańskie, które jak wiadomo, bardzo uważnie obserwują dostawy technologii komputerowej do Europy Wschodniej, nie sprzeciwiają się Waszemu wejściu na rynek radziecki?

– Każda transakcja na dostawę naszego oprogramowania sieciowego do państw Europy Wschodniej posiada licencję eksportową, zaaprobowaną przez amerykańskie organy rządowe. Jestem pewien, że bez tej wstępnej kontroli moglibyśmy sprzedawać 10 razy więcej niż teraz.

– To znaczy, że otrzymanie licencji eksportowej nie jest takie łatwe. Jak długo trwa załatwianie odpowiednich dokumentów?

– Różnie, w zależności od kupującego i od tego, co chce kupić. Nie ma tutaj reguł. Czasami od momentu wysłania odpowiednich dokumentów w USA do otrzymania przez nas licencji mijają tygodnie. Ale zdarzały się przypadki, gdy na odpowiedź trzeba było czekać i pół roku.

– Myślę, że taka sytuacja, mimo ułatwień wprowadzonych przez COCOM w lipcu br. w zakresie eksportu sieci lokalnych bazujących na standardzie Ethernet i Token-ring, pośrednio sprzyja nielegalnemu kopiowaniu Waszego oprogramowania. Tym bardziej, że nadal nie jest jasne, w jakim zakresie ułatwienia w eksporcie sieci odnoszą się do oprogramowania zarządzającego ich pracą.

– Kopie pirackie – to sprawa wywołująca emocje. Mamy z nimi do czynienia przede wszystkim na "młodych" rynkach zbytu oprogramowania. W Stanach Zjednoczonych szczytowy moment emocji związanych z tym zjawiskiem miał miejsce około 10 lat temu, w Wielkiej Brytanii – 5 lat temu. Dzisiaj i my, i inne liczące się firmy produkujące oprogramowanie zrezygnowały z ochrony swoich produktów przed kopiowaniem.

– Dlaczego?

– Jest to niekorzystne dla legalnych, poważnych użytkowników. Taki właśnie wymóg postawił przed nami rynek amerykański, gdzie sprzedajemy ponad 60% naszych sieci. Uważnie obserwujemy sytuację w tej dziedzinie w krajach Europy Wschodniej i doszliśmy do wniosku, że na przykład w Polsce użytkownicy już zaczynają pod tym względem, jeśli można się tak wyrazić, "dojrzewać".

– A jak walczyć z "niedojrzałymi"?

– Tak jak z innymi przejawami kradzieży. Ale w tym przypadku należy zwrócić uwagę na jeszcze jedną okoliczność. Czasami użytkownicy nie wiedzą, że posiadają piracką, nielegalną kopię. Tak więc nie należy z tym problemem przesadzać.

A tak na marginesie: naszym, choć niezamierzonym, sprzymierzeńcem w walce z nielegalnym kopiowaniem są ... wirusy komputerowe, przenoszone najczęściej z bezpłatnym (czytaj "kradzionym") oprogramowaniem.



– Z piractwem oprogramowania, zwłaszcza tak skomplikowanego jak Wasze, można walczyć twórczo. Do tego właśnie zmierzają ośrodki szkolące w obsłudze Waszego oprogramowania. Trzy takie ośrodki działają już w Polsce.

– Doskonale rozumiemy, że użytkownikowi nie może być obojętne, jaką wersję (legalną czy nielegalną) programu posiada. Przecież tylko posiadacz legalnej wersji może liczyć na pomoc w zainstalowaniu sieci, na szkolenie obsługującego ją personelu, na otrzymanie nowych wersji. Jeśli dostawca zapewnia cały kompleks wymienionych przeze mnie usług, problem nielegalnego kopiowania znika.

– Pierwszym krokiem we wskazanym przez Pana kierunku powinno być przygotowanie dokumentacji technicznej w językach narodowych.

– Dokumentacja w języku polskim już powstała. Chciałbym jednak podkreślić, że według kryteriów zachodnich NOVELL jest stosunkowo niewielką firmą. We wszystkich jej filiach, znajdujących się w Kanadzie, Wielkiej Brytanii, RFN, Francji i Hongkongu, pracuje 1500 osób. Nie możemy rozpraszać sił i środków, stąd skupiamy uwagę w pierwszej kolejności na tych rynkach, gdzie sieci NetWare cieszą się największym popytem.

– I ostatnie pytanie. 1992 r. – rok zjednoczenia Europy Zachodniej – już blisko. Data ta będzie przełomowa nie tylko dla polityków. Wiele osób uważa, że właśnie w 1992 r. w Europie rozpocznie się era sieci komputerowych. Ale czy w całej Europie?

– O tym, że rozpoczyna się era sieci komputerowych, słyszę co roku poczynając od 1982 r. Wydaje mi się, że w USA zaczęła się ona przykładowo w 1984 r., w 1985 r. – w Wielkiej Brytanii, trochę później w innych krajach Europy Zachodniej. To oczywiście opinia subiektywna, ale potwierdzona liczbami, ukazującymi skokowy wzrost sprzedaży systemów sieciowych.

W krajach Europy Wschodniej z powodu sztucznych przeszkód w upowszechnianiu technologii komputerowej początek tej ery wciąż przed nami. Ale wszystkie przeszkody stopniowo znikają i 1992 r. faktycznie może być początkiem ery sieci w Europie Wschodniej.

– Dziękuję Panu za rozmowę.



Jan Stożek

Kilka słów o FidoNet

FidoNet to największa na świecie amatorska sieć komputerowa. Specyficzna o tyle, że do założenia węzła nie jest potrzebny specjalistyczny sprzęt – wystarczy komputer klasy PC, modem i dwa, trzy specjalne programy. W porównaniu ze skromnością środków możliwości są zaiste imponujące – wymiana informacji pomiędzy węzłami obu Ameryk, Europy (wraz z Izraelem) i Australii. Tempo jak na możliwości współczesnej telekomunikacji może niezbyt oszałamiające – przekazanie przesyłki do adresata zajmuje 1-2 noce. Należy jednak wziąć pod uwagę fakt, iż zgodnie z projektem przekazywanie korespondencji odbywa się w ciągu zaledwie 1 godz. na dobę (reszta jest przeznaczona na inne cele).

Cały pomysł zasadza się na wykorzystaniu istniejącej sieci telefonicznej wraz z jej infrastrukturą do komunikowania się węzłów. Większość węzłów prowadzi również BBS-y *(Bulletin Board System)* – skrzynki kontaktowe dostępne dla wszystkich posiadaczy komputerów z modemami. BBS służy jako miejsce, w którym wszyscy użytkownicy mogą zostawiać listy do siebie nawzajem, pobierać oprogramowanie, itp. Cała reszta doby poświęcona jest właśnie na pracę BBS-u.

Ponieważ niemożliwe byłoby utrzymanie w ruchu tak ogromnego organizmu bez nadania mu jakiejś struktury, przeto i FidoNet z biegiem czasu, w miarę rozrastania się, obrosła w "sadło" struktury hierarchicznej. Z prostej amorficznej sieci przekształciła się w sieć o prawie ściśle hierarchicznej strukturze.

Podstawowym elementem sieci jest węzeł (Node). Węzłem jest każdy odpowiednio wyposażony komputer wymieniony w NodeList – liście węzłów, jednym z oficjalnych dokumentów FidoNet publikowanym co tydzień w piątki.

Z każdym węzłem związany jest SysOp *(System Operator)*, czyli osoba odpowiedzialna za utrzymanie węzła w ruchu i prowadzenie go w sposób możliwie niekolidujący z innymi węzłami.

Grupa węzłów leżących blisko siebie (na ogół w obrębie miasta lub nawet dzielnicy albo np. województwa) stanowi sieć *(Net)*. Cała komunikacja wewnątrz sieci odbywa się przez bezpośrednie połączenie telefoniczne pomiędzy węzłami, natomiast komunikacja z całą resztą świata przez wyróżniony węzeł zwany patronem *(Most)* sieci.

Sieci leżące w obrębie jednego państwa (w Europie) lub innego "dużego obszaru geograficznego" wraz z węzłami nie należącymi do żadnej sieci (z powodu dużej odległości od innych węzłów) tworzą region *(REGION)*. W przeciwieństwie do sieci region nie pośredniczy w wymianie poczty, natomiast uczestniczy w rozpowszechnianiu oficjalnych dokumentów FidoNet. Zajmuje się też sprawami administracyjnymi w podległym mu obszarze.

Kolejnym stopniem w hierarchii jest strefa *(Zone)*. Strefa odpowiada mniej więcej kontynentowi. Komunikacja wewnątrz strefy odbywa się według zasad przytoczonych powyżej, natomiast pomiędzy strefami przez wyróżnione węzły zwane bramami *(ZoneGate)*. Każda brama zajmuje się wyłącznie przekazywaniem poczty pomiędzy własną strefą, a jedną z pozostałych.

Na czele każdego stopnia hierarchii stoi koordynator *(Net, Region, Zone Coordinator)*. Koordynator odpowiedzialny jest za właściwe prowadzenie podległej mu struktury w sposób najkorzystniejszy dla niej i reszty FidoNet. Koordynator przekazuje informacje o wszelkich zmianach podległej mu struktury do koordynatora wyższego szczebla tak, aby mogły one zostać ujęte w najbliższym wydaniu listy węzłów, i przekazuje aktualną listę węzłów od koordynatora wyższego szczebla do koordynatorów i węzłów niższych szczebli. Koordynator zajmuje się też rozstrzyganiem sporów pomiędzy SysOpami i całym mnóstwem innych rzeczy.

FidoNet ma swoją konstytucję. Jest nią polisa *(Policy)*. Obecnie obowiązuje bardzo kontrowersyjna Polisa 4 (w Europie wariant 4E), trwają jednak prace nad przygotowaniem nowej. Polisa definiuje podstawy struktury sieci, obowiązki i prawa SysOpów, obowiązki i prawa koordynatorów itp. Oprócz polisy obowiązują też normy techniczne – FTSC (*Fido Technical Standard Committee*) określające m.in. protokoły transmisji, format przesyłek transmitowanych przez sieć itp.

Po rozpatrzeniu całej tej struktury nasuwa się prozaiczne pytanie: po co to wszystko? W tym miejscu przypomina mi się anegdota o tym, jak to na jakimś spotkaniu w Moskwie (z okazji jakichś targów czy wystawy komputerowej) prelegent z Zachodu, wybitny gość, po dłuższej prelekcji na temat komputerów i modemów otrzymał pytanie z sali "no dobrze, ale po co to wszystko?". Tu prelegent zająknął się, zamyślił, po czym wydukał: "no, można przesłać program koledze...". Opowiedział mi o tym jeden z redakcyjnych kolegów. Odpowiedziałem mu na to "no, dobrze, a czy mógłbyś wyjaśnić, do czego właściwie służy telefon? Albo długopis?" Odpowiedź wydaje się oczywista, wszak posługujemy się tymi przedmiotami na co dzień, jednak dla kogoś nie znającego ich (lub np. niepiśmiennego) sprawa wcale nie jest taka prosta. Podobnie rzecz się ma z FidoNet. Jest to środek, za pomocą którego ludzie różnych narodowości, mieszkający w różnych punktach globu, mogą się ze sobą komunikować w nieskrępowany sposób zgodnie z własnymi potrzebami. W przeciwieństwie do TV SAT (także docierającej do każdego) FidoNet zapewnia komunikację dwu- i wielostronną, bez niepotrzebnych pośredników modyfikujących przekaz zgodnie z własnym duchem. W przeciwieństwie do zwykłej łączności telefonicznej – nie wymaga, aby odbiorca był w domu, gdy do niego dzwonimy, poza tym jest dużo tańsza – jak zwykle w relacji człowiek – komputer – modem, człowiek jest elementem najwolniejszym. Najbardziej przypomina to skrzyżowanie teleksu (forma przesyłek) z krótkofalarstwem, aczkolwiek skojarzeń takich można znaleźć więcej. FidoNet to forma komunikacji nie wymagająca specjalistycznej wiedzy. Po nauczeniu się stosunkowo prostej obsługi oprogramowania można za pomocą czynności niewiele bardziej skomplikowanych, niż obsługa telefonu czy maszyny do pisania, osiągnąć rezultaty, o których się filozofom nie śniło. ■

```
KOMPUTER

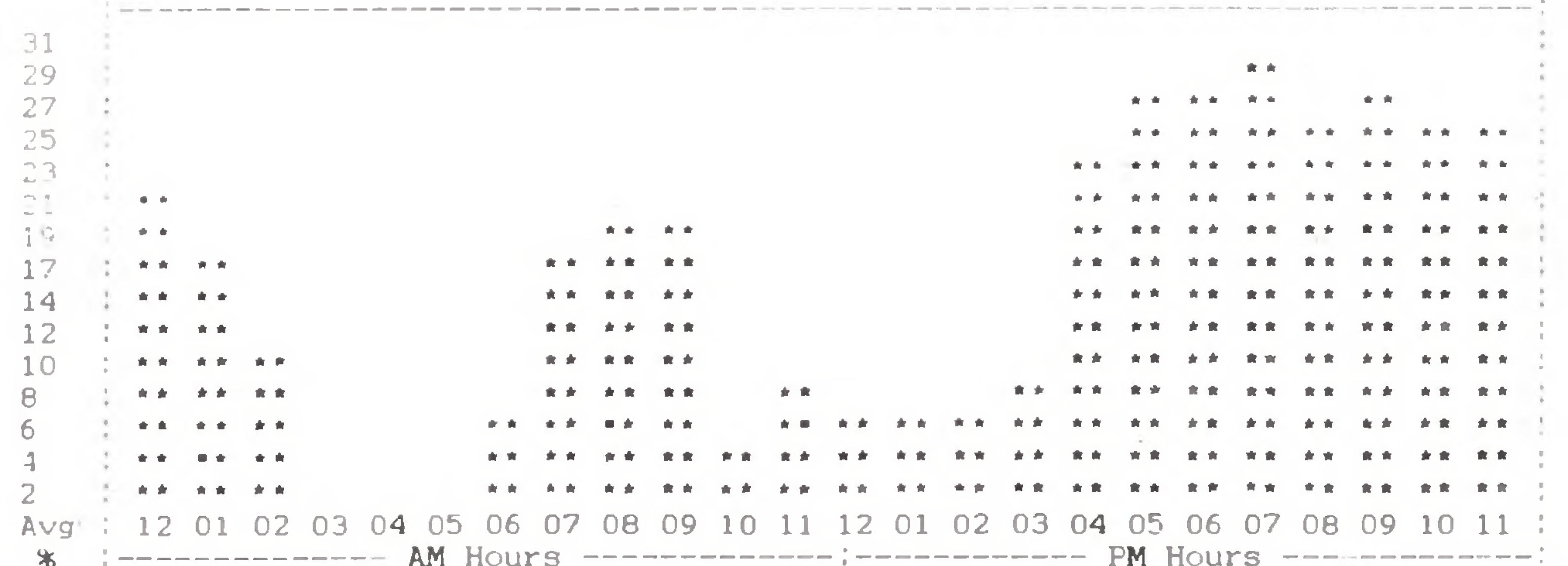
Hot keys available:
    S - Stop any info.
    P - Pause any info.

quick FIDO
```

```
It is Mon 10-Sep-90, 16:25 hr. You have 36 mins left.

<N> - News again              <M> - Mail Submenu
<F> - File Submenu            <K> - KOMPUTER
<E> - EMBBS                   <B> - Bulletines
<C> - Configure QFIDO         <Y> - Yell At SysOp
<V> - Version Info            <A> - Activity Graph
<U> - User List               <S> - Users Statistics
<P> - Play the game           <T> - Time Bank

$ - Your Account              % - Time Statistics
* - TOP Menu                  - - Previous Menu
@ - Letter To SysOp           ! - GoodBye.
```





Równolegle w USA i ZSRR trwa akcja zbierania podpisów pod tekstem poniższej deklaracji, przyjętej w 1989 roku przez Międzynarodowe Forum ds. Własności Intelektualnej (w sferze oprogramowania), które obradowało w Peresławiu-Zalesskim w ZSRR. Akcja ta ma na celu uświadomienie społeczeństwom obu krajów, jak doniosłą rolę odgrywa kwestia intelektualnej własności oprogramowania komputerowego. Stanowisko Polskiego Towarzystwa Informatycznego w zbliżonej kwestii publikowaliśmy w naszym piśmie w ubiegłym roku.

Redakcja

Zasady rozwoju przemysłu oprogramowania

1. Na spotkanie potrzebom gospodarki

Normalny i efektywny rozwój przemysłu oprogramowania odbywa się jedynie w oparciu o zasady mechanizmów rynkowych. Dla krajów, które dopiero zaczynają kształtować swą gospodarkę rynkową, kwestią priorytetową jest budowa wewnętrznego rynku oprogramowania. Zasady funkcjonowania takiego rynku powinny uwzględniać specyfikę narodową, tradycje, opinię społeczną, system ekonomiczny.

Partnerzy zagraniczni powinni szanować specyficzne problemy ekonomiczne krajów o niewymienialnej walucie narodowej i, w miarę możliwości, zgadzać się na rozliczenia w walucie narodowej. Z kolei kraje te powinny sprzyjać tworzeniu warunków, w których partnerowi zagranicznemu będzie się opłacało inwestowanie swych dochodów w walucie narodowej na przykład w rozwój infrastruktury branżowej.

Mechanizm rynkowy powinien zwalczać wszelkie przejawy monopolizacji i niesprawiedliwej konkurencji, z pełnym poszanowaniem praw jednostki i małych firm.

2. Potrzeba przestrzegania praw własności intelektualnej i zasad licencjonowania

Kluczową rolę w kształtowaniu rynku oprogramowania odgrywa pojęcie własności intelektualnej na oprogramowanie, oznaczające konieczność sankcjonowania przez właściciela kopiowania i wykorzystania programu. Zgoda taka powinna być wyrażana w formie umów licencyjnych.

Zarówno doświadczenia praktyczne jak i badania teoretyczne niezbicie dowodzą, iż skutkiem niesankcjonowanego kopiowania jest niska jakość programów, brak usług towarzyszących ich sprzedaży, świadczonych na rzecz ich użytkowników, brak silnych firm softwarowych, brak motywacji do twórczej pracy w tej dziedzinie. Niesankcjonowane kopiowanie programów sprzyja szerzeniu się wirusów komputerowych.

3. Potrzeba nowej etyki i nowego prawa

Najważniejszym warunkiem upowszechnienia się idei własności intelektualnej jest szerzenie nowych zasad etycznych i prawnych. Fundamentalne znaczenie współczesnej koncepcji własności intelektualnej jest skutkiem stopniowego przechodzenia do społeczeństwa informacyjnego. Zaś w związku z trudnościami w wykrywaniu faktów niesankcjonowanego kopiowania programów kluczowym czynnikiem rozwoju przemysłu oprogramowania staje się dobrowolne przestrzeganie norm moralnych, przyjętych w danym społeczeństwie. Do ich przestrzegania powinny zachęcać także czynniki ekonomiczne, na przykład możliwość zastosowania sankcji ekonomicznych wobec jednostek naruszających takie zasady.

Zmiany zasad moralnych zainicjują zmiany w ustawodawstwie narodowym i będą zarazem gwarantować praktyczne przestrzeganie odpowiednich zasad tego ustawodawstwa.

Szczególną rolę w formowaniu nowych zasad moralnych, opartych na przestrzeganiu zasad własności intelektualnej, odgrywa kwestia informowania o konkretnych przypadkach spornych i zapadających w tych sprawach wyrokach.

4. Etyka zawodowa i prawo międzynarodowe

Etyka zawodowa może i powinna zostać umiędzynarodowiona znacznie bardziej, niż normy prawne, regulujące kwestię ochrony praw własności intelektualnej. Zgodnie z Międzynarodową Konwencją Prawa Autorskiego (art. 2) i innymi dokumentami międzynarodowymi przestrzegana powinna być zasada "trybu narodowego", zgodnie z którą autorzy zagraniczni mają takie same prawa, jak narodowi. Jak z tego wynika, jeśli w danym kraju nie są przestrzegane zasady ochrony praw autorskich na rodzime oprogramowanie, jak to jeszcze do niedawna miało miejsce w ZSRR, to brak jest tym samym jakichkolwiek podstaw prawnych do wszelkich roszczeń ze strony firm zachodnich.

Jednakże zasady moralne, przestrzegane przez całą społeczność międzynarodową, wymagają osądzania przypadków niesankcjonowanego kopiowania niezależnie od rozwiązań prawnych, przyjętych w danym kraju.

5. Propagowanie własności intelektualnej: informowanie o zasadach i regulacjach prawnych

Ponieważ koncepcja własności intelektualnej jest stosunkowo nowa, poważną przeszkodą w przestrzeganiu jej zasad jest niedoinformowanie szerokich kręgów ludności, pracujących z wykorzystaniem technik komputerowych. Celem całej społeczności programistów, niezależnie od przynależności narodowej, powinno być jak najszersze upowszechnianie informacji o zasadach, normach, tradycjach, precedensach, ustawodawstwie itd. przyjętych i przestrzeganych przez społeczność międzynarodową. Należy jak najszybciej skończyć z analfabetyzmem w sferze własności intelektualnej. Należy zachęcać do jak najszerszej wymiany informacji w tych kwestiach, sprzyjać kontaktom między ludźmi i organizacjami różnych krajów.

Rynek oprogramowania jest międzynarodowy. Międzynarodowe powinny też być normy i zachowania ludzi i organizacji.

6. Infrastruktura rynku

Niezbędnym warunkiem rozwoju rynku oprogramowania jest kształtowanie odpowiedniej infrastruktury. Podstawowe zadanie przemysłu oprogramowania – to przede wszystkim troska o użytkownika i oferowanie mu nie tylko samego programu, lecz również szerokiej gamy usług – szkolenia, adaptacji programów, usług instalacyjnych.

Rozwój infrastruktury jest ponadto czynnikiem hamującym niesankcjonowane kopiowanie, ponieważ użytkownik zainteresowany jest korzystaniem z dodatkowych usług.

7. Intencje autorów oprogramowania

Rozwinięty rynek oprogramowania oferuje wiele typów stosunków ekonomicznych. Właściciel praw do danego programu może sam wybrać sposób jego rozpowszechniania: bez opłaty (freeware), za dobrowolną opłatą po zapoznaniu się z pracą programu (shareware), kanałami czysto komercyjnymi.

Jednakże granice pomiędzy poszczególnymi sposobami rozpowszechniania powinny być precyzyjnie określone zasadami etycznymi i prawnymi oraz powszechnie znane. Autor powinien sam określać swój zamiar w sposób jawny, na przykład w formie licencji lub innego aktu dołączanego do programu.

8. Apel o działanie

Przemysł oprogramowania jest zjawiskiem międzynarodowym: granice nie stanowią poważnej przeszkody w upowszechnianiu produktów informatycznych, w tym oprogramowania. Ale dla normalnego funkcjonowania handlu międzynarodowego niezbędne jest przestrzeganie ogólnych norm prawnych i moralnych. Odnosi się to zwłaszcza do kwestii ochrony własności intelektualnej.

Zwracamy się do rządów, osób kierujących tą gałęzią przemysłu, biznesmenów, badaczy, wszystkich, którzy wykorzystują i przygotowują oprogramowanie, z apelem o podejmowanie praktycznych kroków, mających na celu tworzenie, upowszechnianie i przestrzeganie odpowiednich zasad. Pierwszym i podstawowym aktem w realizacji tych norm jest osądzanie faktów wykorzystywania programów bez zgody właściciela praw do danego produktu programowego jako bezprawnego i moralnie nie uzasadnionego. Za zgodne z prawem i zasadami moralności powinno uchodzić jedynie wykorzystywanie autorskich wersji oprogramowania (original software). Spełnienie tego wymogu nada przemysłowi oprogramowania silny impuls rozwoju, a postęp w tej dziedzinie ulegnie znacznemu przyspieszeniu.

Każde dobre rozwiązanie zakładające korzystanie z oprogramowania w wersji autorskiej prowadzi do wielu innych pozytywnych konsekwencji, korzystnych nie tylko dla całej gałęzi przemysłu, lecz również dla wszystkich osób w niej zatrudnionych:

- zmniejszenie zagrożenia infekcją wirusami komputerowymi
- podniesienie jakości produkcji
- zapewnienie dobrych jakościowo usług towarzyszących
- budowa rozwiniętej infrastruktury
- możliwość uniknięcia dużych dotacji państwowych (samofinansowanie się tej dziedziny).

Wszelkie teorie lub zasady powinny być realizowane w działaniach praktycznych. Najpełniej odczuwalna realizacja zasady ochrony własności intelektualnej na oprogramowanie – to korzystanie jedynie z autoryzowanych kopii.

Tłumaczenie Haliny Madejczyk



Barbara Włodarek

Muzeum Komputerów

Cały świat szaleje na punkcie miniaturyzacji – sprzęt hi-fi, kamery wideo, kasety, magnetowidy no i oczywiście komputery. Wyjątkiem jest Boston w USA. Chlubą tego miasta jest największy na świecie mikrokomputer. Znajduje się on w muzeum WHARF i zajmuje powierzchnię ponad 100 metrów kwadratowych i dwa piętra. Najciekawsze jest jednak, że ten komputer naprawdę działa. Każdy, a zwłaszcza dzieci, bo głównie z myślą o nich powstał ten olbrzym, może wejść do środka i przekonać się jak on działa. Został zbudowany na wzór i podobieństwo zwykłego domowego komputera łącznie z klawiaturą, monitorem i wszystkimi potrzebnymi układami, połączeniami itp.

Muzeum Komputerów, w którym znajduje się ten olbrzym, powstało w połowie lat osiemdziesiątych. Znaleźć tu można największy zbiór komputerów z lat pięćdziesiątych. Pracownicy muzeum zdecydowali, że nie ograniczą się tylko do pokazywania samych urządzeń. Mówiąc o komputeryzacji nie można przecież zapominać o tym, jaki wywarła ona wpływ na inne dziedziny życia – sztukę, rozrywkę, technikę wojskową, transport itd. Dlatego też specjalny dział poświęcony jest pokazaniu jak wzajemnie wpływają na siebie komputery i inne dziedziny nauki, sztuki i życia.

Chlubą Muzeum jest dział Inteligentnych Maszyn. Dzięki skomplikowanym planszom i objaśnieniom odwiedzający mogą prześledzić rozwój maszyn, które same potrafią "myśleć", czyli można tu zobaczyć najdziwniejsze i całkiem normalne roboty.

Dyrektor dr **Oliver Strimpel** podkreśla, że Muzeum ma uczyć i wpływać na publiczność i to publiczność bardzo zróżnicowaną wiekowo – dzieci i dorosłych. Niekiedy trzeba ich zachęcać, by



przełamali obawy i uprzedzenia, spróbowali dotknąć i pobawić się działającymi eksponatami.

Muzeum mieści się w XIX-wiecznym magazynie. Stare mury i drewniana posadzka są tłem dla nowoczesnej elektroniki. Początkowo, jeszcze w latach siedemdziesiątych, myślano o utworzeniu muzeum, ale dla inżynierów, profesjonalistów zajmujących się komputerami i ich historią. Z czasem ten pomysł uległ zmianie. Postanowiono, że należy raczej pomóc zwykłym ludziom w zrozumieniu tego, co niesie ze sobą komputeryzacja. Odwiedzający Muzeum przechodzą jakby trzy etapy wtajemniczenia. Dowiadują się:

- skąd się wzięły komputery,
- jak działają i co tam jest w środku,
- co potrafią komputery czyli ich zastosowanie.

Zwiedzających najbardziej interesuje ten trzeci etap i dlatego najwięcej miejsca poświęcono pokazaniu zastosowania komputerów w codziennym życiu. Można znaleźć między innymi inteligentne maszyny, a także dowiedzieć się, jakie zastosowanie mają te urządzenia w dwunastu dziedzinach od sztuki i muzyki poczynając, a na organizacji pracy i finansach kończąc.

Po zwiedzeniu Muzeum przeciętny człowiek zaczyna zdawać sobie sprawę z tego, jak bardzo komputery wkroczyły do naszego życia. Właściwie nie ma już chyba dziedziny, w której by nie królowały lub nie pretendowały do miana najważniejszego elementu. Podnosząc słuchawkę telefonu zdajemy sobie sprawę, że gdzieś tam w uzyskaniu połączenia pomaga nam komputer. Kupując bilet na samolot lub pociąg korzystamy z komputera. Powoli do ubezpieczeń różnego typu wkracza komputer. Nawet zakupy można zrobić za pomocą komputera, choć oczywiście nie jest to jeszcze zjawisko powszechne.

Wszystkie te cuda można zobaczyć w jednym miejscu, w Muzeum Komputerów w Bostonie. Dorośli przy wejściu muszą zapłacić 5 dolarów, studenci i uczniowie 4, a dzieci poniżej piątego roku życia wchodzą za darmo. Można też tam kupić książki, plakaty i inne pamiątki związane z komputerami i muzeum. Oprócz tego można nabyć katalog i dwa wydawnictwa – wychodzący co dwa miesiące *"The Computer Muzeum"* i wychodzący raz w roku raport na temat działalności muzeum *"The Computer Muzeum Annual"*. Istnieje możliwość zostania członkiem klubu przyjaciół Muzeum. Wtedy już bez dodatkowej opłaty bierze się udział w różnych imprezach, które odbywają się na terenie Muzeum.

Muzeum Komputerów zwiedza rocznie około 100 tysięcy gości. Dr Strimpel uważa, że jeśli po odwiedzeniu tej placówki choć kilka osób zdecyduje się związać swoją przyszłość z komputerami, ucząc się lub pracując za ich pomocą, to Muzeum odniosło sukces.

\Kurier\

Nowości programowe

De Luxe Animator

Kto na pececie lub Amidze bawił się w grafika ten zna program malarski De Luxe Paint. W lipcu Electronic Arts wypuściła na amerykański rynek w cenie 99 dolarów program standardu **Autodesk Animator** mający wszystkie funkcje animacyjne – **De Luxe Animator**.

Program jest wymagający, minimalne środowisko sprzętowe to IBM PC z pamięcią 640 KB, twardy dysk 20 MB, karta graficzna VGA, monitor VGA i mysz. W pełni możliwości programu wykorzystuje się przy lepszym zestawie – szybkie AT z procesorem 80286 min. 12 MHz, 1 MB pamięci RAM, dysk twardy 40 MB , kolorowy monitor VGA i karta graficzna z trybem VGA/MCGA co daje obraz o rozdzielczości 320x200 punktów i maksymalnie 256 barwach jednocześnie z palety 262144 kolorów.

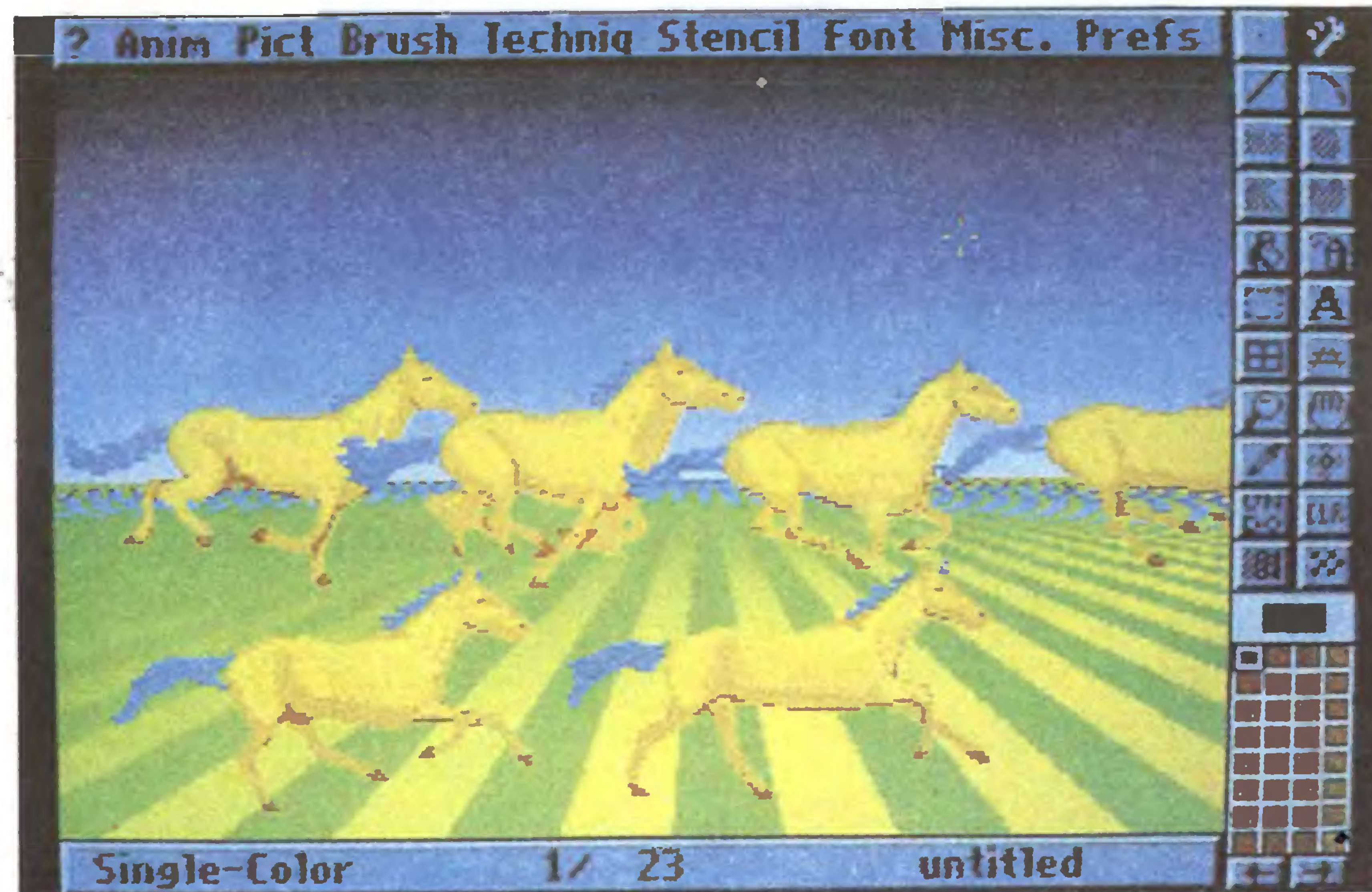
Po zainstalowaniu i uruchomieniu programu otwiera się pole robocze na ekranie monitora. Wybieramy wtedy tryb, w jakim chcemy pracować. Może to być przykładowo 640x480 z 16 kolorami lub 640x200 z 2 barwami itd. Możliwości jest wiele.

Program obsługujemy myszą – bez niej nie możemy wystartować – i klawiaturą. U góry pola roboczego mamy gzyms z rozwijanymi okienkami menu (Pull-Down-Menu), z prawej strony narzędziownie z paletą kolorów.

De Luxe Animator jest programem animacyjnym z pełnym zestawem funkcji malarskich. Może jednocześnie obrabiać 9999 klatek.

Program ma edytor graficzny, animacje dwuwymiarową i trzywymiarową symulowaną przez zmiany perspektywy, efekt zoom i wiele innych wspaniałości. Importuje i eksportuje obrazy w trzech standardach zapisu: LBM (Dpaint), PCX i PCC (PC Paintbrush).

Podręcznik zawiera nie tylko dokładny wykład, jak posługiwać się programem, lecz również podstawowe wiadomości z animacji i montażu filmowego.



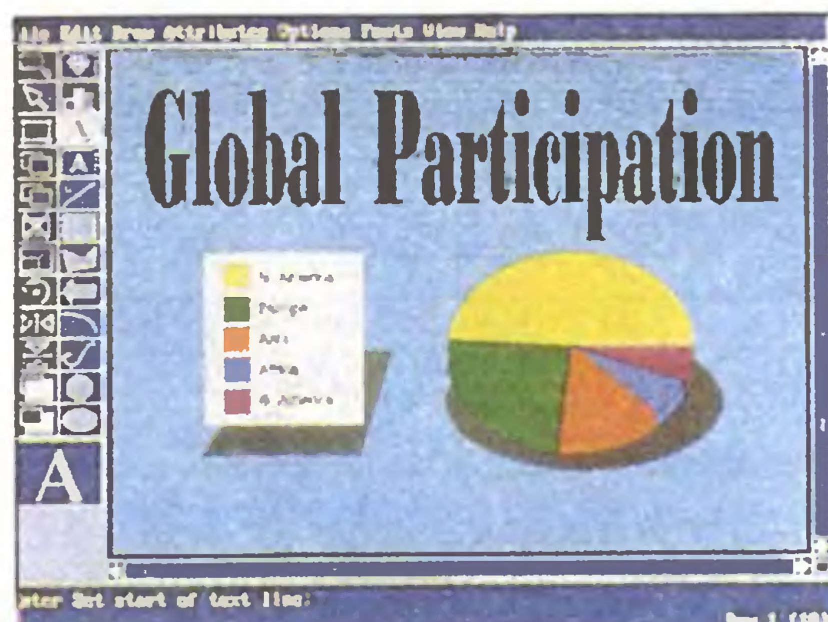
Drawperfect

Drawperfect 1.0 to nowy przebój amerykańskiej firmy znanej na całym świecie z popularnego edytota tekstu Wordperfekt. Jest to program graficzny i prezentacyjny.

Za jego pomocą wzbogacamy nasz referat o kolorowe wykresy i grafiki, które wyświetlamy na ekranie monitora lub dołączamy do opracowywanego tekstu. Drawperfect pracuje w wielu formatach zapisu obrazu: TIFF, PCX, PIC, GEM, CGM, HPGL i WPG.

Jego biblioteka zawiera ponad 500 różnych elementów graficznych. Są to między innymi flagi państwowe, zarysy kontynentów i krajów, zwierzęta, motywy związane z ludźmi i maszynami.

Korzystamy z 25 krojów pism. Możemy też definiować swoje. Dysponujemy zestawem ponad 1500 specjalnych znaków matematycznych i typograficznych. Mamy cyrylicę oraz alfabety grecki i japoński.



Drawperfect przyjmuje obrazy i teksty z Wordperfecta, Lotusa 1-2-3, Excela i Plan Perfecta. Jeśli potrzeba rysujemy "z wolnej ręki". Rozwijane menu i narzędziownie obsługujemy z klawiatury lub myszą. Część programu typu shell jest identyczna z Wordperfectem. Program działa ze wszystkimi kartami graficznymi. Zajmuje 384 KB pamięci roboczej komputera. Wymagany system operacyjny to MS-DOS wersja 2.0 lub wyższa.

Superbase 4

O strukturach i algorytmach baz danych uczą na uczelniach. Jednak obsługa dobrze skonstruowanej bazy danych nie powinna wymagać takiej wiedzy. Te warunki spełnia **Superbase 4**, pracująca w przyjaznym środowisku graficznym Windows.

Pakiet zawiera dziewięć 3,5-calowych lub pięć 1,2 MB 5,25-calowych dyskietek instalacyjnych. Wymaga 2 MB wolnego miejsca na twardym dysku i jeszcze 1 MB dla Run-Time-System MS-Windows. Przy starcie pyta o numer seryjny. Biada tym, którzy go nie znają.

Jednocześnie można pracować w trzech oknach. U dołu każdego jest rząd narysowanych klawiszy, oznaczonych tak jak w domowym magnetowidzie. Służą one do sterowania bazą i są obsługiwane myszą. Porozumiewamy się z bazą również przez okna dialogowe i rozwijane menu.



Dane można importować i eksportować w sześciu formatach: dBase (wersje II, III, III+), Enable, Lotus, Excel, Logistix i DIF (VisiCalc itp.). Możemy też dołączyć rysunki w następujących formatach: IMG, PCX, TIF i WMF (Windows Metafile Format).

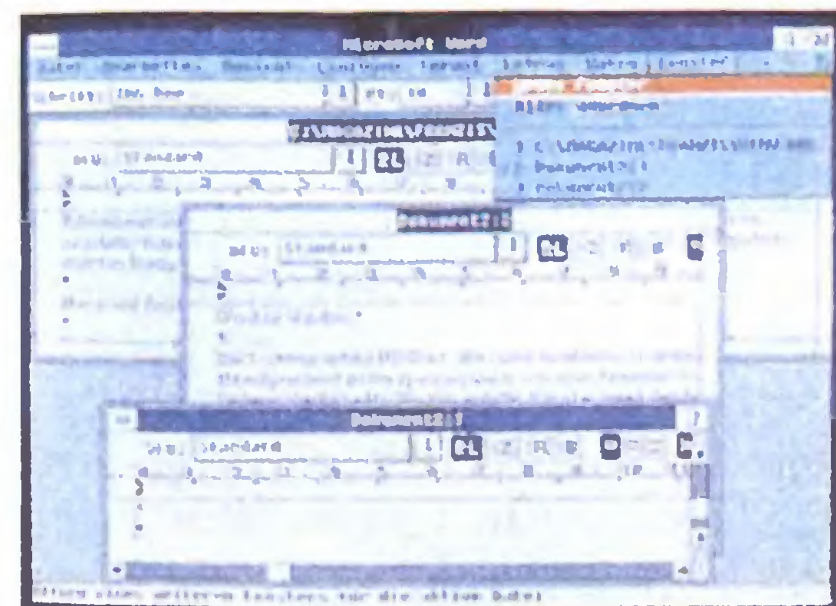
Skomplikowane procedury działań na polach bazy możemy zapisywać w języku podobnym do Basica DML (Database Management Language). Ma on około 280 rozkazów i jest interpreterem.

Dysponujemy również edytorem tekstu podobnym do Write'a. Skomplikowane zależności finansowe zapisujemy jako procedury Macro. Warto więc wydać około 2000 DM, by używać takie cacko.

Niemal DTP

Takim hasłem jest reklamowany znany edytor tekstu **Word** w wersji pracującej w środowisku **Windows**, krótko zwany **Winword** (Win od Windows).

Winword przejął filozofię Windows. Nie przypomina starego Worda 5.0, programista wybrał inną drogę. Wiele różnych okien, rozwijane menu, okienka dialogowe, wszystko obsługiwane myszą.



Można co prawda używać klawiatury, lecz nie jest to wygodne i trudno zapamiętać wszystkie kombinacje klawiszy.

Formatowanie tekstu, użycie różnych krojów pism, podkreślenia, pogrubienia wykonuje się szybko i wygodnie kilkoma tupnięciami myszy. Pomagają w tym dwa małe okienka i spora biblioteka krojów pism i formatów listów, protokołów i innych standardowych dokumentów.

Winword ma wszystkie funkcje dobrego edytora tekstu, między innymi szpaltowanie, nagłówki, stopki, przypisy itd. Jeśli chcemy obejrzeć ostateczną wersję dokumentu przed drukiem wybieramy tryb pracy WYSIWYG.

Dobra funkcja pomocy – help – niezbędna w wielu przypadkach, jest skonstruowana na zasadzie bazy danych MS- Excel. Możemy też zaprogramować często powtarzane ciągi poleceń jako procedury macro.

Winword zaopatrzono w specjalną opcję obróbki informacji. Pozwala ona na wyszukiwanie, sortowanie i wypisywanie danych. Komfortowo rozwiązano tworzenie tabel. Można je również przenosić z baz danych np. z Excel. Można łączyć również tekst z grafiką.

Winword przyjmuje i wydaje teksty w formatach: MS-Word, Wordperfect (wersje 4.1, 4.2, 5.0), Wordstar (wersje 3.3, 3.4, 4.0), IBM PC Text 4, ASCII. Grafiki są importowane w formatach: TIFF, CGM, Lotus PIC, Autocad, Mirage i HPGL.

Ten profesjonalny edytor tekstu wyceniono na 1850 DM.

(JKM)

Kurier

Nowości sprzętowe

Karta graficzna do Megi ST

Użytkownicy Atari ST przyzwyczaili się do pracy z czarno- białym monitorem (bardzo dobrym). Czasem zazdrościli feerii barw amigowcom. Dzisiaj dzięki firmie Maxon Computer otrzymali bardzo dobrą kartę graficzną **MGE II** *(Maxon Graphic Expansion)*.

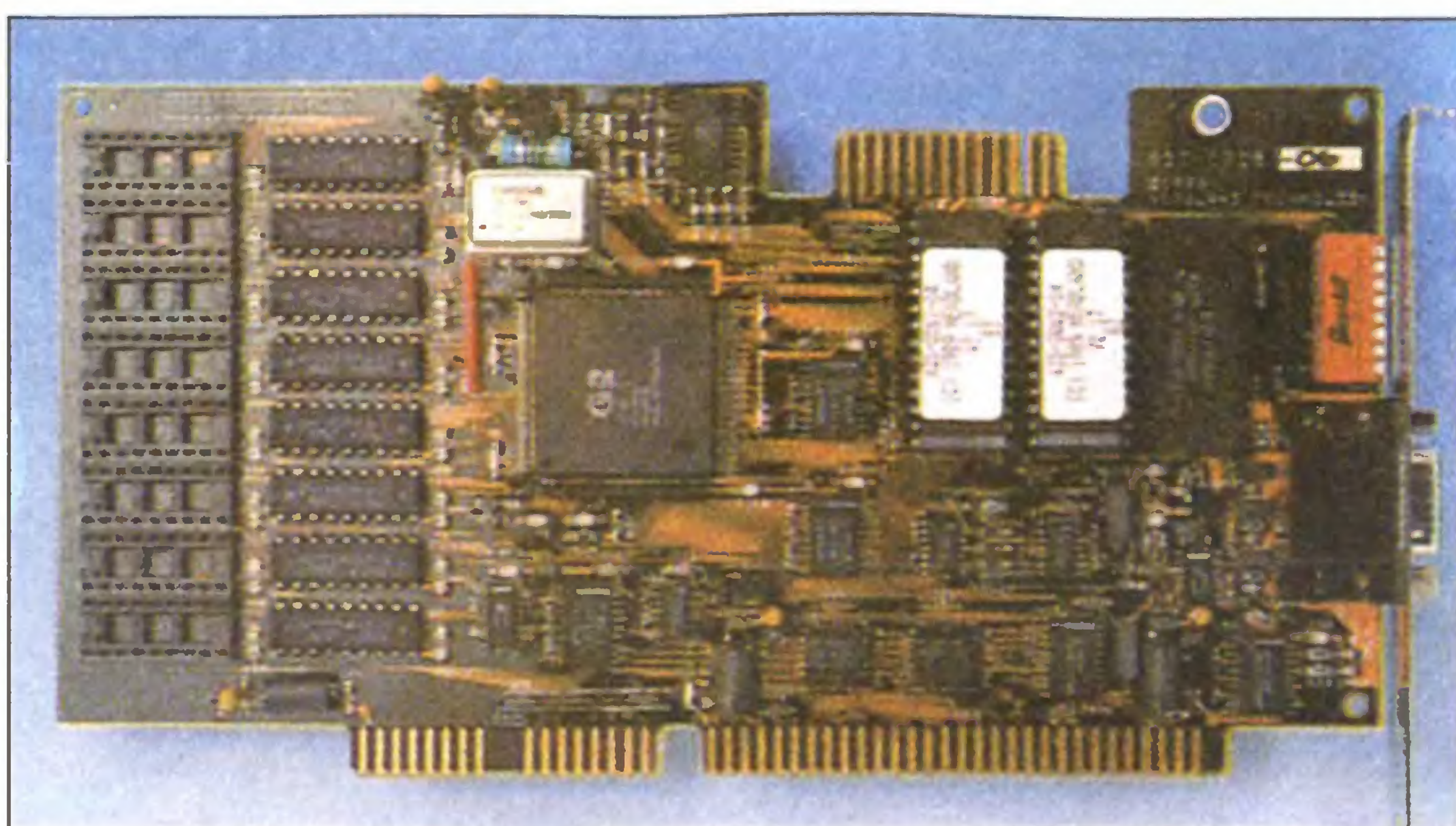
Karta przeznaczona jest do wszystkich wersji **Atari Mega ST**. Jej sercem jest procesor graficzny Intel 82786. Na płytce umieszczono 1 MB pamięci obrazu i wyjścia do monitora VGA, Multisync, ECL. By w pełni wykorzystać możliwości karty polecana jest praca z twardym dyskiem. Kartę umieszcza się na płycie głównej Megi tuż obok procesora. Jeżeli chcemy pracować szybciej musimy kupić koprocesor matematyczny 68881. MGE jest przystosowana do współpracy z nim.

Możemy pracować w siedmiu trybach graficznych: 256 kolorów z palety 262144 barw z rozdzielczością 640x480 punktów (66 Hz) i 800x600 (50 i 80 Hz); 16 kolorów rozdzielczość 896x688 (67 Hz) i 1280x1024 (65 Hz); monochrom rozdzielczość 1280x960 (65 Hz) i 1664x1200 (44 Hz).

Gdy kupimy jeszcze małą płytkę za 50 DM, co jest sumą niewielką przy cenie MGE wynoszącej 2500 marek, i program jej obsługi **Treibersoftware 2.0**, nasza paleta barw wzbogaci się do 16,7 miliona kolorów.

Praktycznie każdy program wykorzystuje grafikę i może pracować z kartą MGE. Maxon automatycznie emuluje standardowy monochromatyczny tryb Atari ST o rozdzielczości 640x400 punktów. Bez problemu uruchomimy znane programy. "Chodzi" nasz ulubiony Signum2! Program DTP DMC Calamus wykorzystuje tryb monochromatyczny 1280x960 punktów, a program malarski Lavadraw 256 barw w rozdzielczości 640x480.

Montaż płytki z MGE II nie jest trudny, choć lepiej zrobić to w autoryzowanym punkcie obsługi. Jeszcze tylko instalacja na twardym dysku lub dyskietce specjalnego programu... i możemy korzystać z różnych egzotycznych monitorów. Otrzymujemy profesjonalną stacje graficzną dla użytkowników DTP, CAD lub Desktop-Video.



Karta graficzna do Windows

23 maja 1990 roku to dzień pojawienia się Windows 3.0. Jednocześnie pojawiło się wiele programów i sprzętu dostosowanego do współpracy z nowym środowiskiem graficznym. Między innymi karta graficzna **VGA 1024i** firmy Video Seven. Nowa karta przeznaczona jest do pracy z Windows 3.0. Pracuje w trzech trybach: 256 kolorów z rozdzielczścią 640x480, 16 kolorów z palety 262144 barw dla 800x600 i 1024x768 punktów. Karta ma też własną specjalną pamięć obrazu V-RAM.

Karty graficzne do TV

Firma Oehlrich & Distler produkuje profesjonalne karty graficzne o standardzie telewizyjnym. **HGY-00** jest w 100 procentach zgodna z normami CCIR (625 linii przy 50 Hz). Ma wyjście sygnału zintegrowany RGB i może być połączona z każdym telewizorem z wejściem SCART lub urządzeniami studyjnymi. Pozwala miksować zewnętrzny sygnał wideo z komputerową grafiką. Wysyła do magnetowidu sygnał w standardzie HQ-PAL-C lub RFU-90.

Obrazy komputerowe mają 800x600 punktów i wykorzystują jednocześnie 256 kolorów z palety 256000 barw. Tryb podwójnego buforowania obrazu (800x300 punktów) pomaga w animacji. Pamięć obrazu typu EMS pozwala wysyłać 25 klatek na sekundę. Specjalny program Screen-Copy przetwarza obrazy TV na standard PC, co pozwala na ich "obróbkę" metodami komputerowymi.

Nowe procesory obrazu

National Semiconductor opracowała trzy nowe 32-bitowe mikrokontrolery do zewnętrznych urządzeń graficznych, takich jak: drukarki całostronicowe, skanery, faksy, modemy do przesyłania danych i terminale. Są to **NS32CG160, NS32FX16** i **NS32GX320**. Wykonano je w technologii CMOS. Pozwalają one na cyfrową, ciągłą obróbkę sygnału, detekcję i korektę błędów. Mogą być montowane w komputerach PC, SUN i VAX. Dołączane jest do nich specjalne oprogramowanie, między innymi biblioteki procedur graficznych, pozwalające wykorzystać ich możliwości.





Super dyskietka

Amerykańska firma **Brier** wypuściła na rynek rewelacyjną dyskietkę 3,5 cala o pojemności 25 MB. Koszt jednej sztuki tylko 25 USD. Nie jest to cena wygórowana za taką pojemność. Ale już około 800 USD więcej musimy wydać, by być właścicielem odpowiedniej stacji. Odczyt i zapis danych odbywa się z szybkością porównywalną z twardymi dyskami.

Płaski komputer

Superpłaski komputer typu *notebook* wyprodukowała firma **Keytronic**. Jest on wielkości kartki formatu A4. Jego serce to procesor Intela 80386 taktowany zegarem o częstotliwości 25 MHz. Pamięć masowa to szybki dysk twardy 20 MB. Płaski ekran ciekłokrystaliczny zgodny z VGA i dobra, wygodna klawiatura – to już standard w tego typu konstrukcjach. Ten ważący 3 kg, zgodny z MS-DOS komputer jest idealnym towarzyszem podróży biznesmena. Keytronic wprowadziła też rewolucyjne rozwiązanie – klawisz zastępujący myszkę. W zależności od naciśnięcia przesuwa kursor nie tylko w górę-dół, ale i w lewo-prawo.

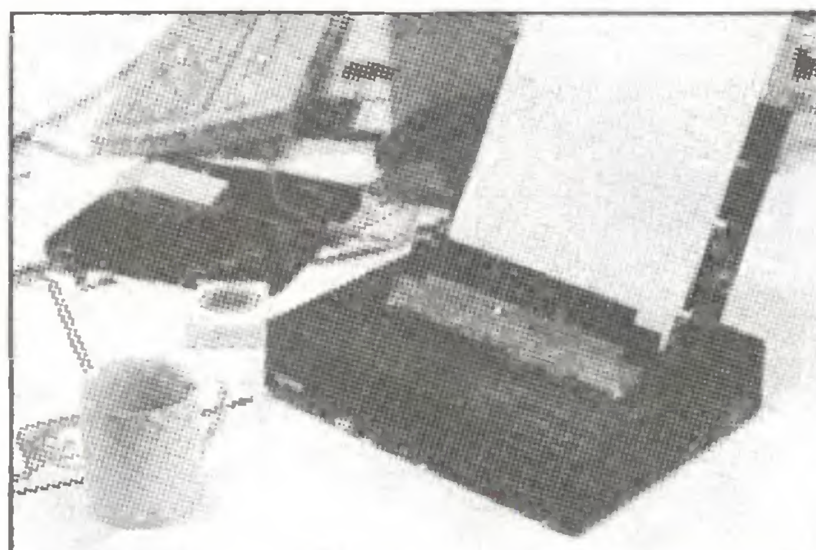
Minilaptop Philipsa

Hasłu "Małe jest piękne" uległa także firma Philips oferując klientom nowy komputer notatnikowy *(notebook)* o nazwie **PLC 101**. Jest to komputer typu XT oparty na procesorze 8086 taktowany zegarem 10 MHz. Wbudowana stacja 3,5 calowych dyskietek ma pojemność 1,44 MB. Ekran pracuje standardowo w trybie Hercules, ale również w CGA i MDA. System operacyjny MS-DOS został "zaszyty" w pamięci ROM. Komputer sprzedawany jest z pakietem programów **Works** wersja 2.0.



Podręczna drukarka

Bogatsi właściciele laptopów mogą za 2500 DM kupić przenośną drukarkę. **Highprint 730** produkowana przez Siemensa jest bateryjną drukarką termiczną. Waży 3,5 kg i drukuje do sześciu stron na minutę (format A4) z rozdzielczością 300 punktów na cal (dpi).



Kolorowy Sharp

Firma Sharp rozpoczęła produkcję przenośnego komputera z barwnym ekranem ciekłokrystalicznym. **Sharp Multi-Color PC-8041** jest wielkości niedużej walizeczki (38,5x39,5x18 cm) i waży 14 kg.

Po rozłożeniu mamy połączoną spiralnym kablem,normalnej wielkości klawiaturę, płaski ekran o regulowanym odchyleniu (25 stopni) i podstawę w kształcie stojącej litery L, w której umieszczono stację 3,5 calowych dyskietek 1,44 MB, twardy dysk 40 MB z czasem dostępu 29 ms, płytę główną i zasilacz. Płaski 14 calowy monitor zgodny z VGA (640x480 punktów) szerokości 28 i wysokości 22 centymetry, ma 307200 punktów podstawowych RGB. Sterowany jest kartą graficzną VGA z pamięcią obrazu o pojemności 256 KB. Paleta barw składa się z 512 kolorów, z tego 16 wyświetlanych jest jednocześnie (max. 256). Kontrastowy i ostry obraz pozwala na długą pracę bez zmęczenia oczu.

Procesor Intel 80386 taktowany zegarem o częstotliwości 8/20 MHz. Pamięć RAM o pojemności 2MB w wersji podstawowej jest rozszerzalna do 8 MB. Pozwala ona korzystać nie tylko z podstawowego systemu MS-DOS 4.01, ale także z Windows 3.0 i OS/2. Komputer ma dwa złącza do kart rozszerzających, złącza RS i Centronics oraz do dołączania zewnętrznych urządzeń: stacji 5,25 cala, monitora VGA, myszy i klawiatury.



Klon Maca

Przyzwyczailiśmy się do różnych komputerów zgodnych z IBM PC, czyli tzw. klonów. Natomiast Mac był synonimem orginału. Oto na rynku amerykańskim, i na razie tylko tam, pojawił się klon Maca firmy Outboud System. Jest on zgodny z **Macintoshem Plus/SE** (wykorzystano orginalny ROM).

Jest to komputer typu *laptop*, wielkości niewielkiego pudełka o wymiarach 31,2x19,8x9,1 cm. Waży z bateriami 4,5 kg. Złożony na kształt zwartego prostopadłościanu.



Do pracy rozkładamy go na dwie części. Pierwsza to klawiatura, druga to podstawa z odchylanym ekranem ciekłokrystalicznym. Płaski ekran ma rozdzielczość 640x400 punktów czyli o 50 procent więcej niż Mac Plus/SE. Zapewnia to lepsze warunki pracy. W podstawę jest wbudowana 3,5 calowa stacja dyskietek. Może ona pracować jako 400, 800 KB i 1,4 MB stacja w formacie Macintosha oraz w formacie IBM jako 720 KB i 1,44 MB. Zamiast stacji montuje się też 2,5 calowy twardy dysk o pojemności 40 MB i czasie dostępu 28 ms.

Łączność między komputerem a klawiaturą jest utrzymywana za pomocą promieni podczerwieni. Klasyczną myszkę zastąpiono urządzeniem zwanym Isopoint. Jest to mały walec, którym obracamy w przód i tył – wymusza to ruchy kursora góra-dół, a przesuwając na boki – w lewo-prawo. Nie jest to łatwe zadanie, gdyż swoboda ruchu wynosi tylko 5 mm. Naciśnięcie wałka jest ekwiwalentem tupnięcia myszą. Konserwatyści mogą podłączyć zwykłą myszkę.

Sercem maszyny jest procesor 68HC000 taktowany zegarem 15 MHz. Standardową pamięć RAM 1 MB można rozszerzyć do 16 MB. Część z niej może być traktowana jako HardRAM i podtrzymywana bateryjnie. Z tyłu obudowy są porty: drukarki, modemu i AppleTalk.

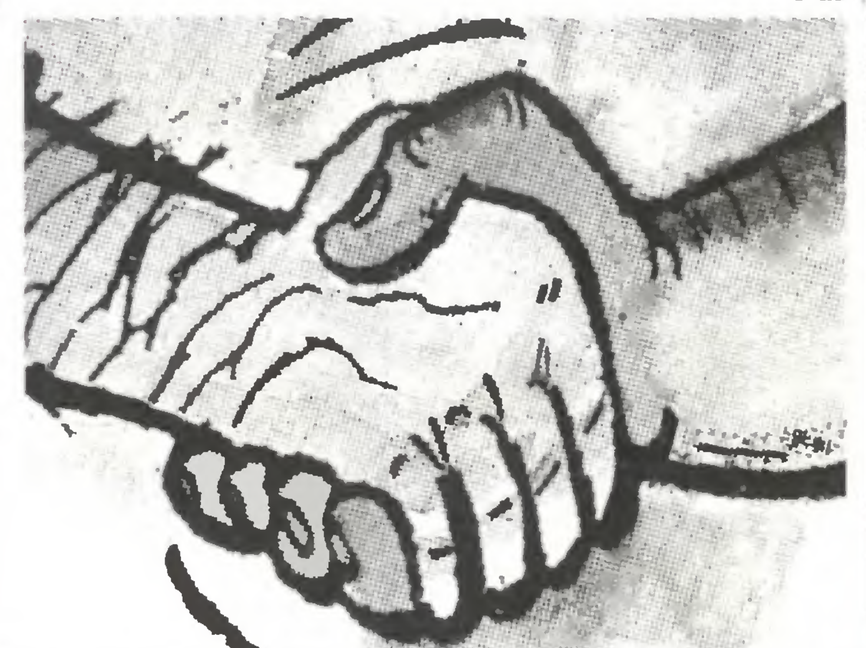
To cudeńko kosztuje około 3000 USD, a z twardym dyskiem 4000.

Nowy Compaq do pracy w sieci

Compaq Deskpro 386N i 286N to dwa nowe komputery przeznaczone do pracy w sieciach.

W 386N umieszczono procesor 80386SX taktowany zegarem 16 MHz, a w 286N procesor 80286 z zegarem 12 MHz. Oba systemy produkowane są w trzech wariantach: Model 0 bez pamięci masowej, Model 1 ze stacją 3,5 calowych dyskietek, Model 40 dodatkowo z 40 MB twardym dyskiem.

Wszystkie modele mają 1 MB pamięć operacyjnej, którą można rozszerzyć do 16 MB dla SX i do 13 dla AT. Komputery pracują w trzech systemach graficznych: VGA, EGA i CGA.



Kurier

Komputeryzujemy się

"Express Wieczorny" w artykule "Trudniej będzie kierowcom uciekać z miejsca wypadku" informuje o komputeryzacji wydziałów komunikacji w stołecznym województwie.

"Komputer był szybszy od mercedesa. Ktoś ze świadków podał numer uciekającego samochodu, milicja korzystając z komputerowej ewidencji ustaliła, kto zacz i gdzie mieszka. Zanim uciekinier zdołał dotrzeć do domu, by zatrzeć ślady uderzenia, już miał niespodziewanych gości. To dopiero zaczątek efektów komputeryzacji ewidencji prowadzonej od dłuższego czasu przez stołeczne wydziały komunikacji. Zakończono pierwszy etap. Komputeryzacja pozostałych wydziałów zakończy się w br. Nie trzeba będzię grzebać w kartotekach ani poza godzinami pracy szukać kierownika i kluczy do nich. Czas po temu najwyższy."

Fakt, najwyższy czas by zacząć komputeryzować naszą policję. Niestety, jak na razie, komputeryzacja ewidencji w wydziałach komunikacji to tylko namiastka. Do tej prawdziwej niezbędny jest komputer centralny gromadzący w swej pamięci wszystkie potrzebne informacje, dostępne w każdej chwili przez modem, w każdym radiowozie. Dzisiaj to tylko marzenia. Brak jest dobrych radiowozów, a co tu marzyć o wyspecjalizowanych komputerowych sieciach. Jak znam naszą rzeczywistość nadal "poza godzinami" trzeba najpierw znaleźć "kierownika", który włączy komputer by na ekranie monitora odczytać szukaną informację. A dopiero później zająć się pościgiem.

* * *

Pozostańmy w kręgu policji. "Słowo Powszechne" donosi o nowym nabytku Instytutu Kryminalistyki w artykule "Komputer identyfikuje przestępców".

"Instytut Kryminalistyki Policji Państwowej w Warszawie od niedawna dysponuje już systemem "portretu pamięciowego", aparaturą "E-FIT" zakupioną w Anglii. Ten przenośny system komputerowy okazuje się nader pomocny przy sporządzaniu portretów pamięciowych sprawców przestępstw na podstawie opisów słownych świadka, dotyczących wyglądu twarzy. Komputer "E-FIT" składa z fragmentów opisu słownego poszczególne elementy twarzy i wyświetla je na monitorze (barwnym), po czym, również korzystając z korekt świadka, modeluje opisaną twarz do granic zapamiętanego podobieństwa osoby poszukiwanej. Kiedy świadek uznaje, że uzyskany obraz odpowiada rzeczywistemu, można monitorowy wizerunek przenieść mechanicznie na arkusz papieru i powielać go w miarę potrzeby, dowolną liczbę razy. Chwilowo policja nasza dysponuje jednym egzemplarzem wspomnianej, przenośnej aparatury."

Za cały komentarz mogłoby posłużyć ostatnie zdanie cytatu. Jednak jeden taki komputer nie powstrzyma wzrostu przestępczości i nie zwiększy wykrywalności sprawców. Być może jesteśmy świadkami początku rzeczywistego doskonalenia metod ścigania przestępców w polskiej policji. Jeżeli tak to jesteśmy za. Jeśli ma to być "kwiatek do kożucha", to nie.

* * *

"Na bladozielonym ekranie monitora ukazują się litery, jednocześnie rozlega się sygnał dźwiękowy i włącza drukarka komputerowa. – "Sklep przy ul. św.Anny – napad – strefa druga, godz. 11.52 data 22.05.90". Kapral obsługujący centralę komputerową wzrusza ramionami. – Na pewno fałszywy. Panienka wierciła się i niechcący wcisnęła przycisk napadowy. Ale sprawdzić trzeba, bo komputer wykluczył awarię."

Brzmi to jak początek kiepskiego kryminału, lecz nim nie jest. Tak zaczyna się artykuł "Bogaty wujaszek" zamieszczony w "Gazecie Krakowskiej", poświęcony systemowi elektronicznej ochrony mienia DGG-16. Idea ochrony sklepów i banków powstała w 1985 roku. System wymyślili, opracowali i skonstruowali pracownicy wydziału łączności WUSW w Krakowie. Dalej czytamy. *"Prace nad stworzeniem systemu trwały około 1,5 roku; pieniądze na ten cel przekazały: PZU – 17 milionów, Spółka Akcyjna Warta – 10 milionów, Urząd Postępu Technicznego – 30 milionów. Resort oficjalnie nie dał ani złotówki. Oficjalnie, gdyż w tym czasie wypłacał pracownikom pensje, płacił delegacje, pokrywał koszty transportu i zaopatrzenia w materiały. Było to możliwe, gdyż opiekunem był sam szef WUSW w Krakowie generał Jerzy Gruba. Resort, czyli MSW, wypłacił także konstruktorom premie za stworzenie DGG-16. W kwietniu 1980 (? red.) roku Sąd Wojewódzki w Krakowie rejestruje spółkę z ograniczoną odpowiedzialnością DYSKAM, której głównymi i jedynymi udziałowcami są: GTS "Wisła" oraz Urząd Postępu Technicznego i Wdrożeń. Założycielami spółki ze strony GTS "Wisła" byli gen. Jerzy Gruba i mjr Marek Połubiński. Kapitał zakładowy spółki wynosi 210 milionów złotych, z czego Urząd Postępu Technicznego i Wdrożeń włożył 100 milionów złotych. Natomiast GTS "Wisła" swój udział określiła na 110 milionów złotych, z czego 1 milion gotówką, maszyny i urządzenia o wartości 51 milionów złotych oraz... myśl techniczną, dokumentację, licencję systemu elektronicznego zabezpieczenia mienia DGG-16. Całe przedsięwzięcie pobłogosławił urzędowo generał Czesław Kiszczak, wydając zezwolenie na działalność spółki ... Nie wiadomo, w jaki sposób GTS "Wisła" stało się właścicielem myśli technicznej, licencji i dokumentacji ... jeżeli i MSW, i klub sportowy oficjalnie nie finansował prac badawczych i kostrukcyjnych. Cały system ... podłączony jest do komputera matki ... który jest własnością resortu, a bez którego cały system DDG-16 nie może pracować, nie działa. Identyczne systemy pracują w kilku miastach Polski: Poznaniu, Olsztynie, Tarnowie, Gliwicach, Warszawie. W jednym przypadku interes prowadzi klub sportowy, w innych spółki detektywistyczne, których pracownikami są w dużej części byli funkcjonariusze MSW."*

Jak widać są i takie aspekty komputeryzacji pracy policji. Z jednej strony to dobrze, że kilka miast w Polsce wzbogaciło się o takie systemy. Być może wpłynie to na ograniczenie włamań i kradzieży. Z drugiej strony sytuacja jest dość niejasna, a przy sprawach związanych z policją tak być nie powinno.

* * *

Zostawmy policję, ale pozostańmy nadal w Krakowie. W "Expressie Wieczornym" przeczytaliśmy:

"Za amerykańskie pieniądze powstaje pod Wawelem system monitoringu komputerowego, którego zamontowanie kosztować będzie równe milion dolarów. Do badania stanu zanieczyszczenia nad miastem posłuży 7 stacji pracujących całą dobę. Wychodzące z nich dane odbierze centralny komputer i z wyprzedzeniem 12 godzin będzie określał potencjalne zagrożenie Krakowa smogiem. Tego nie posiada dotąd żadne miasto w Europie Środkowej. ... Jest to pierwszy etap pomocy amerykańskiej na rzecz ochrony środowiska Krakowa."

Komputerowy monitoring to dobra rzecz. Tylko czy mieszkańcom Krakowa będzie lżej żyć z wiedzą, ile wdychają dwutlenku siarki, tlenków azotu czy węgla i węglowodorów? Lepiej chwalić się czystym powietrzem, niż komputerem do badania ilości szkodliwych substancji.

* * *

W "Przeglądzie Tygodniowym" znaleźliśmy w rozmowie z Andrzejem Ziają, dyrektorem naczelnym "Metronexu", taki oto fragment dotyczący handlu ze Związkiem Radzieckim.

" – Czy przypadkiem nie myśleli także o jakości dostarczanego przez pańską firmę sprzętu? – Najpoważniejszą pozycją są mikrokomputery profesjonalne "Mazovia", jedna z najnowocześniejszych tego typu konstrukcji w krajach RWPG. W ubiegłym roku dostarczyliśmy ich pięć tysięcy, w tym ma być dwa razy więcej. Drugą grupę stanowią drukarki. Najnowsze odmiany wyposażone są w klawiaturę i przystosowane do pracy z mikrokomputerem. Przykładów urządzeń, nie odbiegających zbytnio od standardów europejskich, mógłbym podać znacznie więcej."

Cieszymy się z dobrego samopoczucia Pana Dyrektora, który twierdzi, że sprzedaje *"najnowocześniejsze"* produkty, takie jak "Mazovia". Jesteśmy w stanie to zrozumieć – reklama dźwignią handlu. Chętnie poddamy testowi w redakcji drukarkę, która jest tak nowoczesna, że jest *"przystosowana do pracy z mikrokomputerem"*. Dociekliwych Czytelników informujemy, że wywiad opublikowano 17 czerwca 1990 roku. A mówią, że wokół wszystko się zmienia.

* * *

Na zakończenie kilka zdań z informacji o komputerach "Ze świata nauki i techniki" z czasopisma "Wschód – Zachód".

"Granicą miniaturyzacji jest format A-4. I właśnie takie rozmiary ma najnowszy przebój firmy "Toshiba" – komputer osobisty "Dyna Book" wyposażony w ciekłokrystaliczny ekran o doskonałej rozdzielczości i pamięci, o ogromnej pojemności, większej niż ta, którą jeszcze w połowie lat osiemdziesiątych posiadały duże, stacjonarne maszyny cyfrowe."

Jest to klasyczny przykład informacji, z której niczego nie można się dowiedzieć. Komputer ma *"ekran o doskonałej rozdzielczości"*, pamięć *"o ogromnej pojemności"* itd. itp. Sprostowanie dla szanownych Czytelników. Przebój "Toshiby" to *laptop* T1000SE, a nie "Dynabook" nie będący jej produktem. "Dynabook 286" to komputer przenośny z procesorem 80286, kartą graficzną VGA i pamięcią RAM o pojemności 1 MB, wbudowaną stacją dysków 3,5 cala i twardym dyskiem 20/40 MB. Wniosek nasuwa się sam – lepiej nie pisać niż pisać głupstwa.

(JKM)



Kurier

Czytaj!

Ludwig Classen, Ulrich Oefler "Programowanie systemów mikrokomputerowych", WNT 1989, wyd. I, 7700 + 300 egz., 218 str., seria "Mikrokomputery"

Minęły już lata ośmiobitowych układów mikroprocesorowych i programowalnych układów zewnętrznych, lata fascynacji mikroprocesorami 8080 firmy Intel i Z80 Ziloga. Był to okres, kiedy państwo socjalistyczne starając się dorównać krajom zza "żelaznej kurtyny", podejmowały (z różnym skutkiem) próby produkcji własnych kopii tych najpopularniejszych układów. Nie ominęło to również NRD. U naszych zachodnich sąsiadów wytwarzano rodziny procesorów ośmio-Z80) i szesnastobitowych (Z8000), odpowiednio nazywając je U880 i U8000.

"Programowanie systemów mikrokomputerowych" jest tłumaczeniem książki "Wissenspeicher Mikrorechnerprogrammierung" wydanej przez VEB Verlag Technik (NRD) w Berlinie, w 1986 roku. Poświęcona jest ona programowaniu układów mikroprocesorowych produkowanych w NRD. Są to systemy U880, U8000 i komputer jednoukładowy U881/U882. Sprawy sprzętowe ograniczono do niezbędnego minimum. Więcej miejsca poświęcono programowaniu tych systemów w języku asemblerowym PLZ/ASM. Informacje o programowaniu w Basicu i Pascalu ograniczono do ośmiobitowego systemu U880. Jeden rozdział opisuje operacyjny system projektowo-uruchomieniowy UDOS, wykorzystywany w mikrokomputerach rodziny K1520. W książce zamieszczono wiele rysunków i tabel. Pomagają one w zrozumieniu treści, lecz niestety są mało czytelne. Wydrukowanie książki na marnym papierze IV klasy nie było najlepszym pomysłem. W poszukiwaniu potrzebnego fragmentu tekstu pomaga niewielki, lecz dobrze zbudowany skorowidz.

Przyznam od razu, że nie polecam tej książki. Ukazała się ona ponad dziesięć lat za późno. Tłumaczenie wykonano z II wydania oryginału – rok 1986. Poza tym, jak sami autorzy podają w przedmowie, oparli się na swoich dwóch publikacjach z 1984 roku, a i te nie były pierwszymi wydaniami. Warto przypomnieć, że Z80 pojawił się na rynku w 1977 roku. Być może książka przyda się hobbystom konstruującym swe urządzenia w oparciu o rodzinę Z80 oraz właścicielom Spectrumów, małych Amstradów i MSX-ów, legendarnym procesorem Ziloga, choć z pewnością znajdą lepsze pozycje niż omawiana. Na zakończenie dwa zdania. Pierwsze. Nigdzie nie znalazłem informacji o tym, że U880 to Z80, a U8000 to Z8000 itd.. (Enerdowska nazwa wielu potencjalnym czytelnikom nic nie mówi.) Drugie. Autorzy twierdzą, że przygotowując książkę używali "nowoczesny system redagowania tekstów U880". Niech ten cytat będzie podsumowaniem recenzji.

(JKM)





rys. P. Kakiet



Szanowna Redakcjo!

Zwracam się z uprzejmą prośbą o odpowiedź na następujące pytania:

– Czy w przypadku możliwości zakupu komputera Atari ST wybierać wersje z pamięcią 1 MB, czy wystarczy 0,5 MB – komputer ma służyć do pisania opracowań, tekstów i innych dokumentów, w których oprócz tekstu będą zamieszczane rysunki, szkice (wymiarowane)?

– *Z pamięcią 1 MB.*

– Czy Atari ST poradzi sobie z takimi zadaniami, czy jest to tylko domeną PC/XT/AT?

– *Poradzi sobie lepiej, niż PC//XT.*

– Czy istnieje do Atari ST edytor tekstów równoważny edytorowi ChiWriter 2.04, który świetnie się nadaje do pisania tekstów i tworzenia rysunków w tekście?

– *Są edytory znacznie lepsze.*

– Jaka jest możliwość uzyskania polskich liter na takim edytorze do ST i ich wydruk na drukarce (posiadam drukarkę Brother M1209)?

– *Polskie litery ładowane są programowo.*

– Jak wygląda naprawdę sprawa potrzeby instalowania dwóch monitorów do Atari ST – kolorowego i czarno-białego – czy całe oprogramowanie nie może "chodzić" z monitorem kolorowym?

– *Poważne oprogramowanie „chodzi" z monitorem monochromatycznym.*

– Jak Redakcja sądzi, czy Atari ST może być komputerem w miarę nowoczesnym przez najbliższe lata (stosowany będzie głównie w domu) – czy istnieje możliwość połączenia wyników pracy (tekstów, baz danych, arkuszy kalkulacyjnych itp.) z komputerem standardu PC/XT?

– *Tekstów i grafiki – tak.*

– Jak wygląda praca na ST z oprogramowaniem wykorzystywanym na PC/XT – czy jest możliwość pełnej i poprawnej emulacji – na jakiej drodze taka emulacja jest realizowana (dodatkowe przystawki czy programowo)?

– *Programowe emulatory PC//XT*

– Jak wygląda możliwość współpracy ST ze stacją 5,25 cala, czy "chodzą" wszystkie programy – jaka jest pojemność dyskietek? – słyszałem różne opinie na ten temat.

– *Współpracuje bez problemów. Pojemność dyskietkach zależy od zastosowania stacji.*

– Czy w przypadku stosowania tylko monitora kolorowego należy kupić typ SC 1224, czy można stosować obecnie produkowane przez Polkolor telewizory z wejściem monitorowym, czy może to być monitor np. Philips?

– *Można stosować telewizory z wejściem monitorowym.*

– Czy jest w Polsce jakiś Klub lub Koło Użytkowników ST, gdzie można uzupełnić wszelkie wiadomości w dziedzinie oprogramowania ST?

– *Klubów jest wiele. Warszawski działa przy Politechnice (patrz niżej).*

– Jakie są dostępne pozycje literatury w języku polskim na temat Atari ST i gdzie można je nabyć?

Bardzo proszę o rzetelną informację na temat poruszonych przeze mnie problemów. Pozwoli mi to podjąć decyzję w sprawie wyboru tzw. docelowego komputera domowego. Waham się pomiędzy Atari ST a PC/XT. Mam również dzieci, które czasami chcą trochę pograć w gry komputerowe.

Jarosław Krajewski
Pruszków

Od redakcji: O Atari ST pisaliśmy i piszemy dużo (rubryka STragan). Wszystkim szukającym literatury poświęconej komputerom polecamy warszawską księgarnię, przy ul. Mokotowskiej 51/53 tel. 28 16 14, nad którą redakcja objęła patronat.

* * *

Jestem stałym i wiernym czytelnikiem Waszego miesięcznika. Spowodowane to jest głównie tym, że od 16.02.1987 r. obsługuję pierwszy w naszym zakładzie mikrokomputer klasy IBM PC/XT. Pracuję w Hucie Małapanew w Ozimku k/Opola w dziale Głównego Energetyka.

Obecnie w naszej hucie działa już kilka mikrokomputerów (w tym jeden oryginalny IBM) i w trakcie wdrażania są 3 sieci lokalne. Od pewnego czasu zainstalowane są również duże jednostki typu Odra i Riad (nie znam ich klasy).

Jeżeli chodzi o moje bezpośrednie "podwórko" wygląda to niewesoło. Od momentu zainstalowania sprzętu oprócz mnie nikt się tym nie zajamuje i nikt też nie został przeszkolony w jego obsłudze.

Po powrocie z urlopu oczekuje

na mnie stos dokumentacji, którą muszę pośpiesznie wprowadzić, aby uaktualnić bazę danych (silniki elektryczne). Cały wolny czas w pracy jak również po pracy przez 2 lata poświęcałem na naukę i szkolenie się w informatyce (samodzielnie). Dzięki temu zapoznałem się nieźle z GWBasicem, częściowo z Turbo Basicem, a obecnie rozgryzam dBase III Plus wraz z kompilatorem Clipper 86.

W naszej hucie jest specjalny Dział Informatyki, w którym pracuje między innymi kilka osób zajmujących się projektowaniem i programowaniem profesjonalnym na dużych maszynach i mikrokomputerach. Ja z moimi problemami jestem traktowany raczej jako intruz zabierający im cenny czas. Za pośrednictwem spółek zewnętrznych piszą programy na mikrokomputery dla Huty i innych zakładów w okolicy.

Zainteresowanie rozwojem informatyki i komputeryzacji jest w naszej Hucie raczej znikome (oprócz gier zręcznościowych). Oferowałem czasowy dostęp do mojego mikrokomputera (jest jednostką autonomiczną) różnym zainteresowanym, lecz gdy przyszło do stwierdzenia, iż to wymaga nauki, osoby te rezygnowały.

W tym roku byłem kolejny raz na winobraniu w RFN i wróciłem stamtąd jako właściciel XT Turbo 4,77 MHz, z 20 MB dyskiem twardym 1x360 KB, monitorem mono 14 cali, wyposażonym w HGC oraz drukarki LC-10. Jak na amatora to chyba nie jest zły zestaw, tylko teraz jest problem, jak go prawidłowo wykorzystać.

Gdybym był "pełnowartościowym" programistą to w Opolu miałbym szansę znaleźć dodatkową pracę w jednej ze spółek informatycznych. Szukam obecnie możliwości praktycznego przeszkolenia się w znajomości Clippera, lecz jak dotychczas bezskutecznie.

Teraz zapoznaję się z możliwościami drukarki i rozpocząłem oczywiście od przeczytania artykułu pana Zenona Rudaka z nr 2/88 o tejże drukarce tylko kolorowej.

Ten tekst próbuję napisać przy pomocy spolszczonej wersji edytora ChiWriter, gdyż na codzień używam NE.

Informatycy dotychczas zakupili jedynie 2 programy kalkulacyjne polskiego pochodzenia (a ja używam niemieckiej wersji MultiPlanu), a reszta to są kopie zagranicznych programów narzędziowych. Oczywiście Huta płaci bardzo drogo za konkretne programy użytkowe różnym firmom.

Mam więc prośbę i zapytania do panów zajmujących się sprzętem w Waszej redakcji.

1. Nie wiem jak prawidłowo powinien być konserwowany sprzęt komputerowy. Ja osobiście dokonuję co miesiąc czyszczenia zewnętrznego przy pomocy flanelowej szmatki, a jeden z informatyków odkurza wewnątrz i czyści napędy dysków elastycznych przy pomocy specjalnej dyskietki. Chętnie zapoznałbym się z instrukcją napisaną po polsku, jak taka konserwacja winna być prawidłowo przeprowadzana.

2. Podczas transportu w RFN ekran mojego monitora uderzył dosyć mocno o obudowę jednostki centralnej. Nie wiem czy to jest przyczyną, lecz obecnie w górnej części monitora (pas szerokości ok. 3 cm) widać wyraźnie odchylenie od pionu (zewnętrzne) i poziomu (górne) linii na ekranie. Szczególnie widoczne było to przy testowaniu za pomocą programów Benchmark, chociaż koło graficzne w części środkowej było prawie idealne. Nie wiem czy istnieje możliwość usunięcia tej wady.

3. Podczas pracy mojego domowego komputera nastąpił wybuch typowy dla zwarć w instalacji elektrycznej. Okazało się, że eksplodowało ogniwo zasilające zegar czasu rzeczywistego. Obecnie muszę przy każdej inicjacji systemu wprowadzać aktualny czas i datę. Na szczęście wszystkie układy scalone pozostały na kartach rozszerzających (krótkich) w porządku.

Czy jest możliwość zakupu i montażu takiego ogniwa i co mogło być przyczyną jego zwarcia?

Na te pytania nie potrafię sobie sam odpowiedzieć i nie wiem do kogo się z nimi zwrócić.

Byłbym wdzięczny za kilka zdań merytorycznej odpowiedzi. Podczas czytania tekstów informatycznych, testów czy reportaży z zagranicy zamieszczanych w "Komputerze" zazdroszczę nieraz, że nie mam możliwości kontaktu osobistego z jednym z takich profesjonalistów.

Z informatycznym pozdrowieniem

M. Tokarek

Od redakcji: Przedstawiony w liście obraz komputeryzacji przedsiębiorstwa rzeczywiście nie jest wesoły. Mamy nadzieję, że zmiany zachodzące w kraju spowodują również zmiany w zakładzie pracy. W sprawie monitora i ogniwa należy zwrócić się do najbliższego serwisu. Wymiana samego ogniwa może zlikwidować skutek, a nie przyczynę. Nie polecamy samodzielnych napraw. O konserwacji sprzętu napiszemy w jednym z kolejnych numerów "Komputera".

* * *

Szanowny Panie Redaktorze,

Chcę zabrać głos na temat szeroko poruszany w Waszym (a w zasadzie naszym) piśmie, dotyczący znaków polskiego alfabetu w komputerach.

W programach typowo obliczeniowych polskie znaki w reprezentacji liczb nie występują, więc praktycznie nie mają znaczenia ich kody. Natomiast w programach przetwarzających teksty kody są bardzo istotne. Do takich programów możemy zaliczyć wszystkie aplikacje pisane w systemie Clipper lub dBase. Załóżmy, że w bazie danych występuje pole znakowe NAZWISKO. W tym polu mogą znajdować się nazwiska np. pracowników jakiegoś zakładu. Dodatkowo załóżmy, że wprowadzono indeks w oparciu o pole NAZWISKO. Zamysłem programisty oraz oczekiwaniem użytkownika jest, aby układ alfabetyczny przy przeglądaniu kartoteki (narzucony przez plik indeksowy) był zachowany również dla polskich liter. Będzie to najbliższe ideału w przypadku stosowania kodów polskich liter według Domu Handlowego Informatyki (dawny Dom Handlowy Nauki Polskiej). Co prawda wszystkie nazwiska zaczynające się od polskich znaków zostaną umieszczone na końcu, ale relacje porządkujące zostaną zachowane. Wynika to z faktu, że Ą<Ć, Ć<Ę,... ą>Ż, ą<ć itd. Relacje te nie są spełnione w przypadku pozostałych dwóch standardów kodów polskich znaków (Latin 2 i Mazovia). Podobne problemy wystąpią przy pisaniu programów w języku C czy Pascal, o ile w programach używane będą operacje porównywania łańcuchów. Tak więc popieranie np. standardu Latin 2 wymaga głębszego zastanowienia, bo kto przystosuje kompilatory i inne narzędzia programisty?

Z poważaniem

Antoni Starczynowski
Olkusz

Od redakcji: My przyjeliśmy standard Mazovii. A jak radzą sobie Czytelnicy? Chętnie zamieścimy Wasze wypowiedzi.

* * *

Droga Redakcjo!

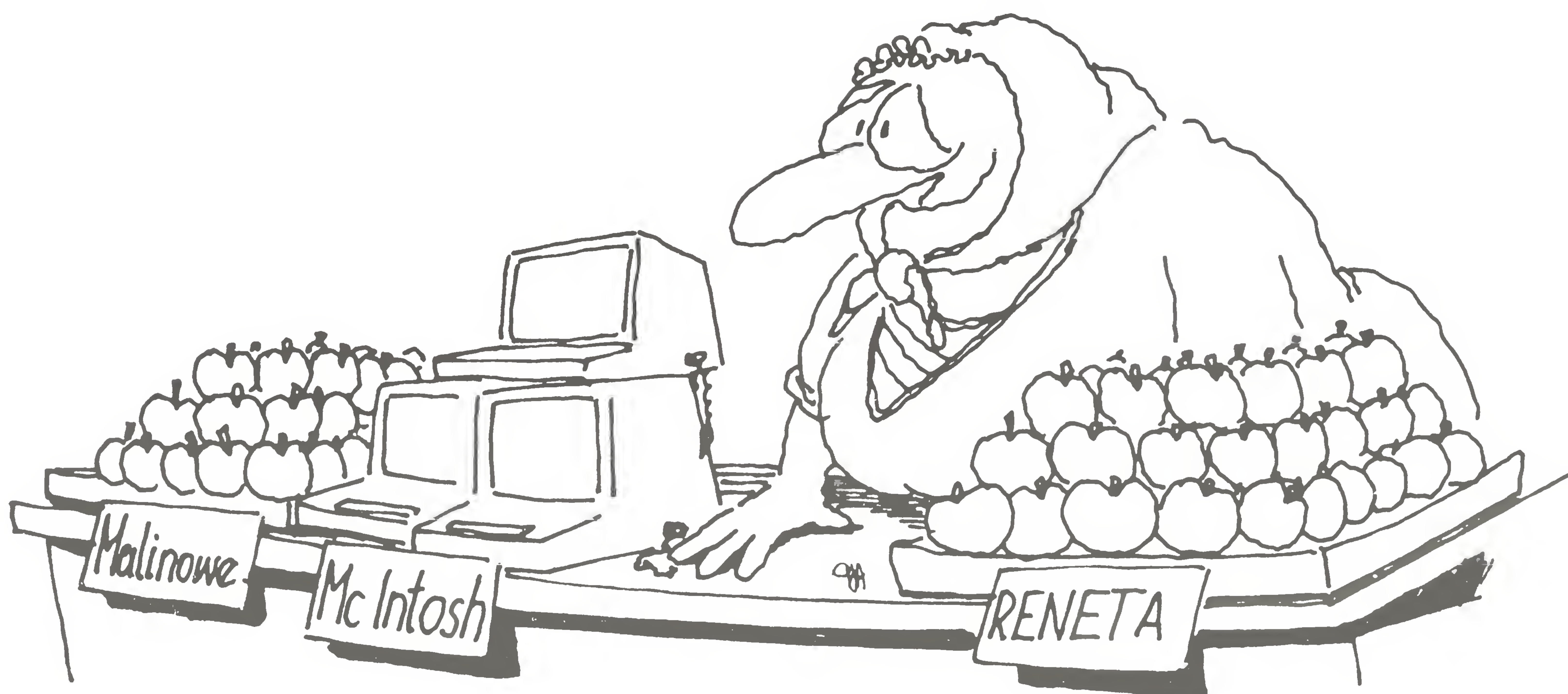
Mam 19 lat i od pewnego czasu posiadam Atari 1040 STF z monitorem SC 1224. Chciałbym w pełni wykorzystać możliwości tego zestawu, ale niestety brak mi jakiejkolwiek literatury na temat Atari ST. Proszę o informacje o zasadach funkcjonowania i przynależności do Klubu Użytkowników ST działającego przy Waszej redakcji. Mam nadzieję, że dzięki Wam uzyskam dostęp do podręczników i ciekawych programów.

Ireneusz Jakóbik
Szczecin

Od redakcji: Klub ST już od dłuższego czasu nie działa przy naszej redakcji. Niestety nie wiemy kiedy i gdzie będą odbywać się klubowe spotkania po wakacjach. Dotychczas klubowicze spotykali się w każdy drugi piątek miesiąca, w gmachu Wydziału Elektroniki Politechniki Warszawskiej.



rys. P. Kakiet



Andrzej Urbankowski

„Nieśmiertelność" na dyskietce

W artykule "Życie – Energia – Czas" (Komputer nr 5 i 6/90) przyjęto założenie, że poszukujemy "usprawnień" do gier w wersji tasmowej. Gra taka stanowi z reguły jeden plik informacji wprowadzanej do pamięci komputera. Poza nielicznymi wyjątkami po uruchomieniu gry i wykonaniu RESET można uruchomic ją ponownie po odszukaniu adresu startowego programu co umozliwia pracę "włamywaczowi". Posiadacze stacji dysków 1541 doskonale wiedzą, że programy gier "dyskietkowych" składają się z kilku segmentow, które wgrywają się samoczynnie zależnie od postępow w grze, a także automatycznie uruchamiają. Jest faktem, że w niektórych bardziej rozbudowanych grach programista uprzejmie zapytuje nas, czy życzymy sobie byc "niesmiertelni". Nie jest to jednak regułą, a ciągłe oglądanie komunikatu Game Over staje się denerwujące. Czy jesteśmy bez szans na zmianę sytuacji? Nic podobnego. Przy grach stosunkowo prostych istnieje możliwosc, że po jej wgraniu i wykonaniu RESET grę uda się uruchomić powtórnie, co sprowadzi zagadnienie na już nam znany grunt.

Wyszukiwanie "usprawnień" do gier na dyskietkach zacząłem od gry **Breakout'80**. Kto z nas nie rozbijał kulką muru z cegieł?! Adres startowy odnajdujemy bez trudu, jako $0980. Mamy 5 kulek do dyspozycji, więc metodą jak w grze **Commando** znajdujemy:

```
13E4 LDA#$05
13E6 STA$CF22
```

Natomiast rozkaz DEC$CF22 występuje pod adresami: $17BO, $20CC, $2408, $2494, $8501, $8E1D, $9159, $91E5. Jest ich osiem, czyli tyle ile poziomów gry. Zastąpienie ich rozkazami LDA$CF22 ułatwi nam doprowadzenie dzieła demolowania muru do końca. To było banalnie proste, ale powoli wprowadza nas w specyfikę pracy ze stacją dyskow.

Druga gra, ktorą się zajmowałem, to **Staff of Karnath**. Uznałem ją za wartą zaprezentowania z uwagi na nietypową wersję energii, którą dysponuje bohater. Jest to energia wyrażona w procentach. Na początku gry wynosi ona 100%. Wygląda to lepiej niż 100 "żyć", ale i tak okazuje się, że jest jej za mało, gdyż każdy strzał do różnych stworów pilnujących części amuletu uszczupla jej zapas. Nie mówiąc o zetknięciu się z nimi! Tę grę też rozpracowano metodą **Commando**, czyli:

```
0A70 LDA#$64 ;64hex 100
0A72 STA$6FFE
1602 DEC$6FFE
```

Zastąpienie rozkazu DEC$6FFE rozkazem LDA$6FFE jest skuteczne o tyle, że licznik energii kręci się w kołko od 0 do 100, ale gra się nie kończy. Mozna go oczywiście zatrzymac, tak aby cały czas wskazywał 100%. Nie jest to trudne i odszukanie tej bardziej eleganckiej wersji poprawki do gry, którą uruchamia SAS$0A00, pozostawiam Czytelnikowi.

Kolejna gra, z którą nie poszło już tak łatwo, to **The Goonies**. Limit 8 "żyć" to zdecydowanie za mało, aby zdobyć skarb Jednookiego. Uruchomienie gry po RESET nie udaje się. Nie załamujemy jednak rąk, lecz zabieramy się za przegląd zawartosci dyskietki. Wciskamy

F7 i po chwili widać, że gra składa się z jednego tytułowego segmentu o długości 167 bloków (1 blok to 256 bajtów) oraz dziesięciu segmentów o długości od kilku do kilkunastu bloków, stanowiących zapewne kolejne etapy gry.

Przypominamy sobie, że główny blok gry nie uruchamiał się samoczynnie, ale dopiero po RUN następowała dekompresja programu oraz wgranie obrazka tytułowego. Nasza szansa tkwi w tym, że być może procedura kontrolująca limit "żyć" zawarta jest w bloku głównym gry, a następnie przez "przeładowywacz" jest przenoszona we właściwe miejsce pamięci. Wgrywamy blok główny i przechodzimy do monitora. Znów jest niezawodna metoda **Commando**. Przy piątej próbie znajdujemy:

233F LDA#$08
2341 STA$12B7
0AFD DEC$12B7

i, jak zwykle, zastępujemy DEC przez LDA, co daje oczekiwany efekt. Jeszcze chwilę zajmie nam odszukanie adresu startowego $083C dla bloku głównego, co jest konieczne, gdyż po wyjściu z monitora i użyciu OLD oraz RUN program nie chce się uruchamiać. Po wprowadzonej zmianie i uruchomieniu bloku głównego (a w konsekwencji całej gry) przez SYS$083C ujrzenie komunikatu o uroczystym przyjęciu w poczet Goonies'ów jest tylko kwestią czasu i pomysłowości gracza.

Omówione przykłady stanowiły jedynie wstęp do problemu wyszukiwania i wprowadzania zmian w grach na dyskietkach. Chcę to omówić na przykładzie gry **Airborne Ranger** – najlepszej symulacji pola walki, jaką dotychczas widziałem. Gra ta, typowo militarna, niewiele ma wspólnego z bezmyślną strzelaniną. Fragmenty programu gry są nakładkowane, wgrywają i uruchamiają się samoczynnie, a próba odnalezienia adresu startowego zakończyła się niepowodzeniem (może ktoś z Czytelników dokona tej sztuki?). Czarna rozpacz, ale nie załamujmy się. Spokojnie obserwujemy proces wgrywania się gry. Najpierw czołówka, potem obrazek, drugi obrazek, podobizny autorów, dłuższy okres wgrywania się programu i wreszcie pojawia się plansza **Assignement**, a program czeka na wybór przez nas kilku istotnych dla gry danych. Zamiast je podawać wciskamy FREEZE i sekwencją F1-F7-F1 przechodzimy do monitora. Przeglądamy pamięć i stwierdzamy, ża zajęte są komórki $0800 do $1230 oraz $4C00 do $6796. Wracamy do Basica i próbujemy SYS$4C00. Pojawienie się po chwili planszy Assignement, poza satysfakcją, dało nam informację o stanie pamięci (jej zajęcia) w momencie uruchamiania gry. Pozwala to przewidywać, że właściwy program gry umieszczony będzie od adresu $8000. Wybieramy misję dla naszego bohatera i wgrywamy grę do końca. Po pojawieniu się planszy **Prepare to deploy** znów wciskamy FREEZE i przechodzimy do monitora. Zgodnie z przewidywaniami od $8000 do $F000 pojawił się program gry, zniknął program umieszczony poprzednio od $4C00 do $6796, natomiast nienaruszony pozostał program zajmujący komórki $0800 do $1230. No dobrze, zapyta zapewne bardziej niecierpliwy Czytelnik, po co to wszystko, skoro i tak nie potrafimy odnaleźć adresu startowego?! Tak, to prawda. Możemy jednak zmienić program bezpośrednio na dyskietce! Wprowadzenie zmienionego programu do pamięci komputera również zaowocuje różnymi "usprawnieniami". Jak tego dokonać? Sam **Final II** tu nie wystarczy. Potrzebny jest program narzędziowy znajdujący na dyskietce rozkaz assemblera, który nas interesuje. Przez znalezienie należy tu rozumieć odszukanie numerów ścieżki i sektora, w których występuje. Niestety, nie dysponowałem takim programem, a próby jego pozyskania wśród znajomych oraz na zlotach użytkowników Commodore spełzły na niczym. Napisanie go w assemblerze, aczkolwiek możliwe, wymagałoby sporej pracy, a ponadto okazało się, że zdobycie wydruków zawartości pamięci ROM w stacji dysków komputera wraz z komentarzem (bez tych materiałów napisanie programu jest praktycznie niemożliwe) jest jeszcze trudniejsze niż znalezienie gotowego programu. Pozostał więc Basic i wykorzystanie procedur bezpośredniego dostępu do dyskietki. Program taki został napisany przez autora i w wersji skompilowanej przeszukuje całą dyskietkę w czasie około godziny. Tak długi czas przeszukiwania dyskietki bardzo utrudnia pracę i dlatego pozwolę sobie na pewien partykularyzm. Jeżeli ktoś z Czytelników ma taki program w assemblerze, proszę o kontakt za pośrednictwem Redakcji. Ze swej strony chętnie udostępnię zainteresowanym swój.

Pozostawmy jednak problem narzędzi i wróćmy do gry. Musimy teraz wykonać kopię dyskietki z grą, gdyż nieumiejętnie przeprowadzone zmiany na oryginale mogą doprowadzić do jego zniszczenia. Do prób wystarczy skopiować tylko stronę A dyskietki. Teraz usuwamy z niej programy MISA1, MISS1, MISA2, MISS2, MISA3, MISS3 pozostawiając resztę, a wśród nich programy MISA0 i MISS0, odpowiadające misji **Disable SAM Site**. Dzięki temu mamy pewność, że odszukane ścieżki i sektory oraz wprowadzone zmiany cały czas dotyczą jednej i tej samej misji. Dlaczego jest to takie ważne nie trzeba chyba uzasadniać. Wgrywamy misję Disable SAM Site i po uruchomieniu gry przechodzimy do monitora. Analizujemy program, począwszy od adresu $8000. Napotykamy ciąg podprogramów. Przypuszczamy, że niektóre z nich "zarządzają" fragmentami programu, które stanowią przedmiot naszego zainteresowania. Autor zbadał kilka z nich, uzyskując ciekawe rezultaty. I tak:

1) JSR$8870 – obecność bunkrów
2) JSR$85F8 – obecność pól minowych
3) JSR$E800 – strzelanie z bunkrów
4) JSR$E802 – strzelanie przez żołnierzy
5) JSR$E89D – strzelanie z pułapek

W praktyce należy sprawdzić, jaki rozkaz znajduje się pod adresem wywoływanym przez podprogram, a następnie używając monitora dyskowego odszukać ścieżkę i sektor, na których on występuje. Aby mieć pewność, że odszukana

> 16



15 <

komenda (a raczej jej lokalizacja) dotyczy "naszego" rozkazu, należy również uwzględnić bajt przed nim i po nim. Najlepiej wyjaśni to przykład. Załóżmy, że podprogram, który chcemy wyłączyć, zaczyna się od adresu 8630. Po disasemblacji widzimy:

```
862F  60        RTS
8630  A9 00     LDA#$00
8632  8D 23 82  STA$8223
8635  A2 04     LDX#$04
```

Musimy więc znaleźć ścieżkę i sektor, w których znajduje się sekwencja bajtów 60 A9 00 8D lub też (co jest nawet pewniejsze) 8D 23 82 A2. To jednak jeszcze nie wszystko. Teraz korzystamy z możliwości manipulacji zawartością sektorów, jakie daje **Final II**. W monitorze wpisujemy *R 06 0A i wciskamy RETURN. Spowodowało to, że wczytaliśmy do pamięci od adresu $CF00 do $CFFF dziesiąty sektor z szóstej ścieżki. Możemy teraz, wykorzystując wbudowany w Final II assembler, dokonać dowolnych zmian wgranego sektora, a więc również wpisać RTS w adres wywoływany przez podprogram. Operacje takie przeprowadzaliśmy już wielokrotnie. Jednak przy dokonywaniu takiej operacji na dyskietce adres, pod który mamy wpisać RTS, będzie inny niż w pamięci komputera. Tak więc nie $8630 a np. $CFA1. Ustalić to można z łatwością przeszukując pamięć od $CF00 do $CFFF na obecność poszukiwanej wcześniej sekwencji bajtów, wykorzystując opcję H (Hunt). Jeżeli teraz, po dokonaniu zmiany, wpiszemy *W 06 0A i wciśniemy RETURN, to do dziesiątego sektora na szóstej ścieżce zapiszemy jego zmodyfikowaną wersję. W opisany sposób można zmieniać program nie w pamięci komputera, a na dyskietce, obserwując jednocześnie efekt wprowadzanych zmian. Dla pięciu przebadanych podprogramów odszukałem następujące ścieżki i sektory oraz adresy pamięci, w których występujący tam rozkaz należy zastąpić przez RTS:

1) 06 05 CF84 LDA#$00
2) 06 04 CF08 LDA#$08
3) 07 11 CF14 LDY#$00
4) 07 11 CF16 STX$E871
5) 07 11 CFB1 LDA#$00

Wyznaczenie ich dla innych misji pozostawiam Czytelnikom.

Pozostał także do znalezienia licznik błędów. Nie musi on wcale być umieszczany w programie wgrywanej misji, ale może znajdować się w obszarze $0800 do $1230. Niezależnie od lokalizacji jego odnalezienie wymagać będzie uruchomienia wyobraźni i wsparcia jej intuicją, aby zminimalizować czas poszukiwań. Będą tu pomocne rozważania zamieszczone w "Życie – Energia – Czas".

Pozostała jeszcze kwestia gier zabezpieczonych przed kopiowaniem z dyskietki na dyskietkę. Starsza wersja gry **Airborne Ranger** takich zabezpieczeń nie ma i odczytywanie danych bezpośrednio z dyskietki przez monitor dyskowy nie stanowiło problemu. Nowa wersja tej gry ma już takie zabezpieczenia. Ich usunięcie wymaga wiedzy znacznie większej, niż można się spodziewać u początkującego "włamywacza". Jednak problem ten można obejść nie kopiując całej dyskietki, ale kolejne, już wgrane do pamięci komputera misje. Wykorzystujemy do tego celu przycisk FREEZE. Przeszukanie dyskietki, na którą "zrzucono" w ten sposób dowolną misję, nie napotyka już trudności. Jest to metoda pachnąca piractwem, ale skuteczna! Niech usprawiedliwi nas fakt, że robimy to nie dla zysku, ale dla przyjemności.

I jeszcze parę słów na temat tzw. niepublikowanych rozkazów mikroprocesorów rodziny 65xx. Dobrze jest wiedzieć, że rozkazy takie istnieją, warto też znać ich możliwości. Nie należy jednak przeceniać ich roli. W swojej praktyce autor ani razu nie zetknął się z rozkazem tego rodzaju.

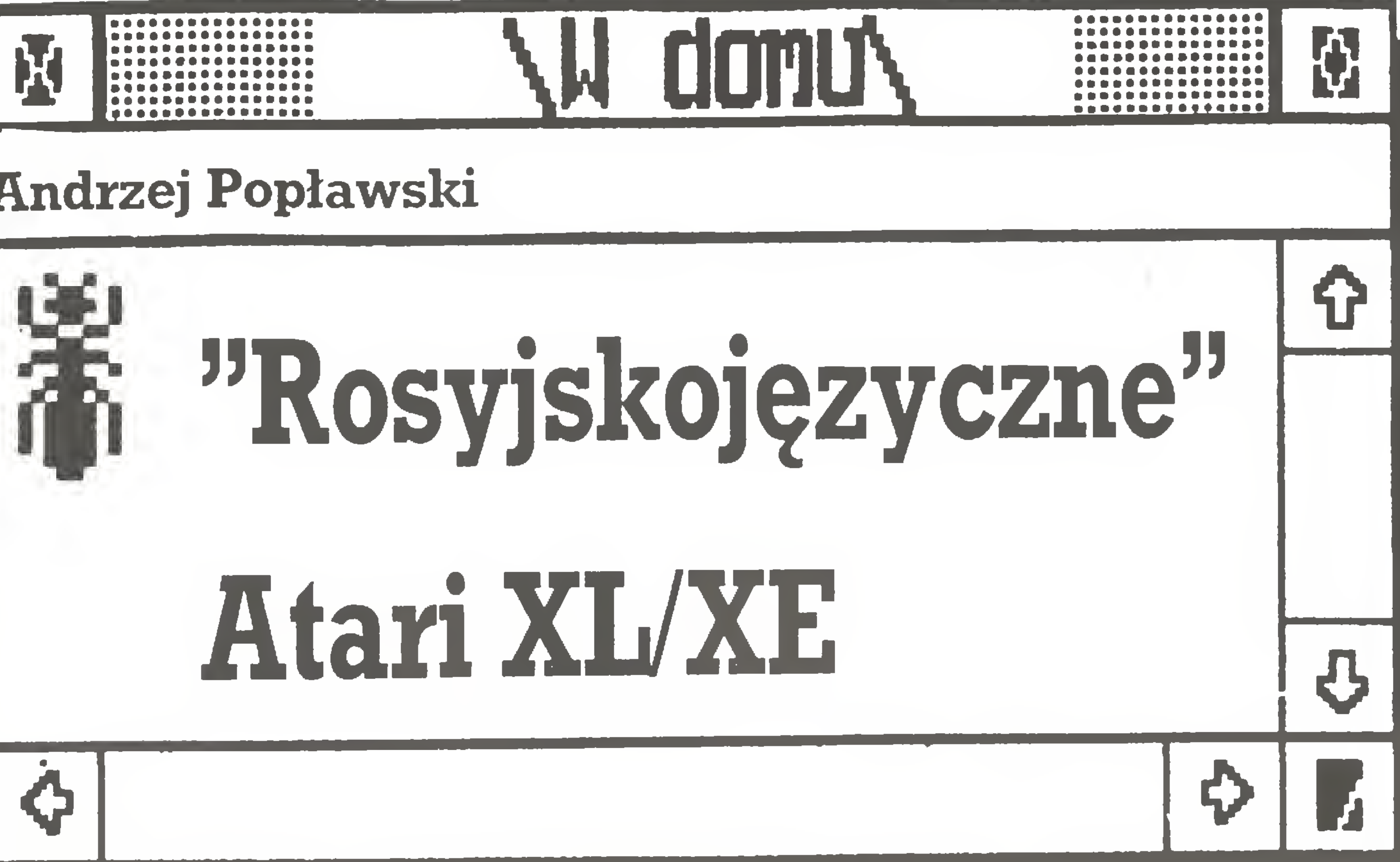
Intencją tego materiału jest zachęcenie do podjęcia samodzielnej pracy przez Czytelników, zwłaszcza tych początkujących w assemblerze, którzy chcieliby połączyć przyjemne z pożytecznym. Tak, jest to możliwe! Na początek sukcesem będzie znalezienie "nieśmiertelności", ale później przyjemność sprawiać będzie zatrzymanie upływu czasu lub energii, aż wreszcie zaczniesz szukać, Czytelniku, możliwości wprowadzania poprawek w sposób bardziej elegancki bądź też inną metodą. A kiedy napiszesz swój pierwszy program w assemblerze, z uśmiechem przypomnisz sobie, jak to wszystko się zaczęło. Nie można oczywiście zakładać, że ktoś biegle opanuje assembler bez literatury i sporego wkładu pracy, zajmując się jedynie "włamaniami" do gier. Ale stopniowo, jeżeli zechce, może taką znajomość osiągnąć. Myślę, że problem trafnie ujmuje przysłowie: dla chcącego – nic trudnego. Autor nie jest profesjonalnym programistą, a 3 lata temu nie odróżniał bitu od bajtu z tej prostej przyczyny, że nie wiedział o ich istnieniu. Więc – można! Dzisiaj mogę dodać, że warto!

Pisząc cykl poświęcony poprawkom do gier starałem się uniknąć zbędnego teoretyzowania, co, jak mam nadzieję, będzie przychylnie ocenione przez Czytelników.

Temat wprowadzania poprawek do gier nie po raz pierwszy pojawia się na łamach "Komputera". Zainicjował go cykl "I ty zostaniesz włamywaczem" autorstwa G. Czapkiewicza.

Swoje doświadczenia z "włamań" do gier autor postanowił przekazać głównie z uwagi na fakt, że filozofia mikroprocesorów rodziny 65xx, a więc i ich język maszynowy, znacznie się różnią od popularnego Z 80, co zapewne zniechęciło wielu Czytelników do podejmowania prób "włamań".

Jeżeli cykl opracowań mojego autorstwa sprawi, że ilość "włamań" do programów dla komputerów C64 i Atari wzrośnie, będę mógł uznać, że czas poświęcony na ich napisanie nie był czasem straconym.



Andrzej Popławski

"Rosyjskojęzyczne" Atari XL/XE

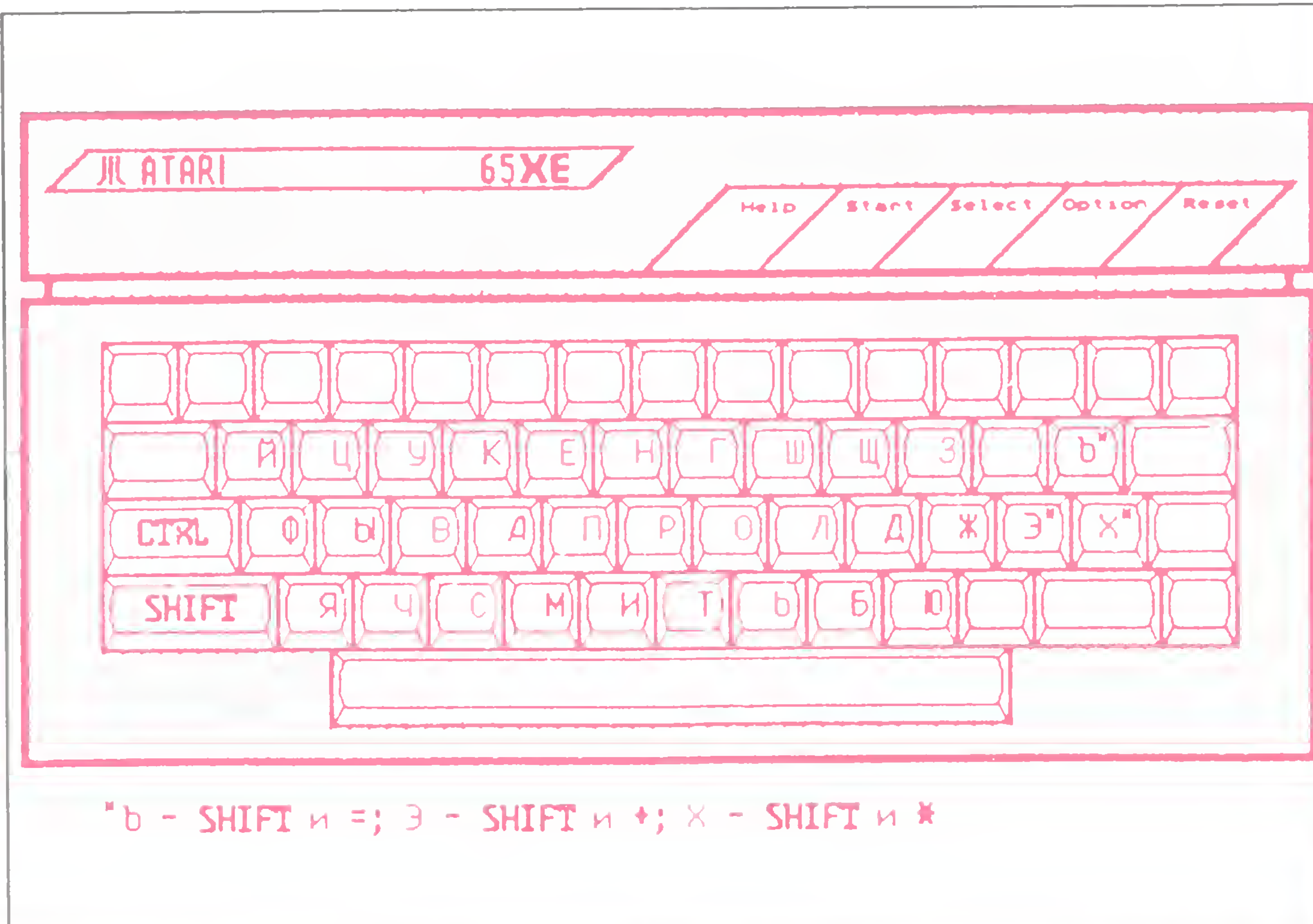
Małe Atari zajmuje wśród "ośmiobitowców" drugie miejsce – po Spectrum – pod względem popularności w ZSRR. Obecnie, mimo wszelkich trudności i problemów, w tym kraju lawinowo rośnie zainteresowanie komputerami i informatyką jako nowym hobby. Prawdopodobnie przemiany w Europie Wschodniej zaowocują również wzrostem kontaktów między komputerowcami z obu stron granicy na Bugu. W związku z tym przydadzą się rosyjskie literki dla Atari XL/XE. Proponowana procedura pozwala pisać komentarze do programu po rosyjsku.

Przypomnijmy w skrócie informacje, niezbędne do nauczenia naszego Atari rosyjskiego alfabetu. Generator znaków Atari składa się ze 128 znaków. Tabela znaków zajmuje w ROMie przestrzeń między adresami 57344 i 58367: od 57344 znaki specjalne, interpunkcyjne i cyfry; od 57600 – wielkie litery; od 57856 – znaki graficzne i od 58112 – małe litery. Razem 1024 bajty (128 znaków po 8 bajtów każdy). Istotne są adresy 756 i 54281: tu przechowywany jest adres początku tabeli znaków, a dokładniej numer strony pamięci. I jeszcze jeden adres – komórka pamięci 106, przechowująca ilość stron pamięci, którymi może dysponować BASIC.

Przejdźmy do tekstu programu. Wiersz 29000 określa nową liczbę dostępnych stron pamięci i nowy adres tabeli znaków. Wiersz 29010 – przepisanie tabeli generatora znaków w nowe miejsce i poinformowanie interpretera, od którego wiersza programu ma czytać dane. Wiersz 29020: czytanie adresu znaku. Przy adresie równym -1 następuje przełączenie na nową tabelę znaków i powrót do programu głównego. Wiersz 29030 – czytanie danych znaków i przenoszenie ich do nowej tabeli. Wiersze 30000-30031 zawierają opisy kształtu rosyjskich liter w następującej kolejności:

Układ liter jest prawie zgodny z radzieckim standardem klawiatury (standard – przeniesiona jest tylko litera). Na ekranie rosyjskie litery pojawiają się przy jednoczesnym naciśnięciu CONTROL i odpowiedniego klawisza (rosyjskimi literami zostają zastąpione właśnie te znaki, które oryginalnie uzyskuje się takim sposobem). Wyjątek od tej reguły stanowią litery , i , które uzyskujemy odpowiednio poprzez naciśnięcie SHIFT i odpowiednio =, + i *. Rosyjskie litery można umieścić również inaczej, zmieniając ich adresy, czyli pierwsze liczby w liniach 30000-30031. Pozostałe 8 wartości w każdym wierszu opisują kształty znaków (a więc też można je swobodnie zmieniać).

Proponowaną procedurę można "wklepać" i uruchomić jako samodzielny program, ale lepiej zapisać na taśmę lub dysk instrukcją LIST i następnie dołączyć do potrzebnego programu instrukcją ENTER. Budowa programu będzie następująca: wiersz 5 – wywołanie podprogramu z rosyjskimi literkami, dalej program główny, zaś na końcu wiersze 29000-30032.





Cyrylica dla Spectrum

Władysław Kramarienko

Proponowany program przeznaczony jest dla ZX Spectrum. Kody wprowadzanych do pamięci komputera rosyjskich znaków odpowiadają radzieckiemu standardowi KOI-7.

Tabela znaków zajmuje 768 bajtów i może być umieszczona w wolnej części RAM. Jeżeli RAMTOP ustawiony jest poniżej początkowego adresu obszaru zajmowanego przez nowe znaki, to instrukcja NEW nie skasuje ich z pamięci. RAMTOP ustawiany jest poprzez instrukcję CLEAR xxx, gdzie xxx=początkowy adres tabeli kodów minus 1.

Wiersze 5-60 – to procedura ładująca (loader), zamieniająca wartości heksadecymalne z wierszy 9903-9998 (tu znajdują się właściwe dane) na wartości dziesiętne. W każdym wierszu danych po instrukcji REM znajduje się znak, w miejsce którego po uruchomieniu programu wejdzie znak o nowym kształcie, opisywanym w tym wierszu.

Po wpisaniu tekstu programu wydajemy polecenie RUN, a następnie LIST – dla sprawdzenia kształtu nowych znaków. W razie wykrycia błędów w ich kształcie poprawiamy odpowiadające im wiersze programu – i znowu RUN. Tabelę znaków zapisujemy na taśmę instrukcją **SAVE "R.ALF" CODE 64000,768.**

Odczyt z taśmy – instrukcja **LOAD "R.ALF" CODE** lub **LOAD "R.ALF" CODE n**, gdzie n – nowy adres, podzielny przez 256. Przełączanie z rosyjskiego alfabetu na łaciński dokonuje się przez **POKE 23607,60**, z łacińskiego na rosyjski – **POKE 23607,x**, gdzie x=adres, podzielony przez 256, minus 1.

Tłumaczenie: Andrzej Popławski

```
  1 REM R.ALPHABET
  3 CLEAR 63999
  5 LET A=10: LET B=11: LET
C=12
 10 LET d=13: LET e=14: LET
f=15
 12 LET x=64000: POKE
23607,249
 15 READ a$
 20 FOR n=1 TO 15 STEP 2
 30 LET w=VAL a$(n)
 40 LET w=w*16: LET
w=w+VAL a$(n+1)
 50 POKE x,w: LET x=x+1
 60 NEXT n: GO TO 15
```

```
9903 DATA "0000000000000000":
REM " "
9904 DATA "1818181818001800":
REM!
9905 DATA "6C6C000000000000":
REM "
9906 DATA"6C6CFE6CFE6C6C0
0": REM #
9907 DATA "187EE07C0EFC3800":
REM $
9908 DATA "00C6CC183066C600":
REM %
9909 DATA"386C3877DECE7B00":
REM &
9910 DATA "1818300000000000":
REM '
9911 DATA "000C181818180C00":
REM (
9912 DATA "0030181818183000":
REM )
9913 DATA "00663CFF3C660000":
REM *
9914 DATA "0018187E18180000":
REM +
9915 DATA "0000000018183000":
REM ,
9916 DATA "0000007E00000000":
REM -
9917 DATA "0000000000181800":
REM .
9918 DATA "03060C183060C000":
REM /
9919 DATA"7CC6CED6E6C67C00":
REM 0
9920 DATA "1838181818183C00":
REM 1
9921 DATA "3C66061C30667E00":
REM 2
9922 DATA "3C66061C06663C00"
: REM 3
9923 DATA "0E1E36667F060F00":
REM 4
9924 DATA "7E607C0606663C00":
REM 5
9925 DATA "1C30607C66663C00":
REM 6
9926 DATA "7E66060C18181800":
REM 7
9927 DATA "3C66663C66663C00":
REM 8
9928 DATA "3C66663E060C3800":
REM 9
9929 DATA "0018180018180000":
REM :
9930 DATA "0018180018183000":
REM ;
9931 DATA "0C18306030180C00":
REM <
9932 DATA "00007E00007E0000":
REM =
9933 DATA "6030180C18306000":
REM >
9934 DATA "3C66060C18001800":
REM ?
9935 DATA "0000CEDBFBDBCE
00": REM @
9936 DATA "00003C063E663F00":
REM A
9937 DATA "043C607C66663C00":
REM B
9938 DATA "0000666666667F03":
REM C
9939 DATA "00007E666666FFC3":
REM D
9940 DATA "00003C667E603E00":
REM E
9941 DATA "00187EDBDBDB7E18":
REM F
9942 DATA "00007C6060606000":
REM G
9943 DATA "0000C66C386CC600":
REM H
9944 DATA "000066666E766600":
REM I
9945 DATA "180066666E766600":
REM J
9946 DATA "0000666C78666600":
REM K
9947 DATA "00000E1E3666C600":
REM L
9948 DATA "0000C6EED6C6C600":
REM M
9949 DATA "000066667E666600":
REM N
9950 DATA "00003C6666663C00":
REM O
9951 DATA "00007E6666666600":
REM P
9952 DATA "00003E663E366600":
REM Q
9953 DATA "00007C667C606000":
REM R
9954 DATA "00003C6660663C00":
REM S
9955 DATA "00007E1818181800":
REM T
9956 DATA "00006666663E063C":
REM U
9957 DATA "0000DB7E187EDB00":
REM V
9958 DATA "00007C667C637E00":
REM W
9959 DATA "000060607C667C00":
REM X
9960 DATA "0000C6C6F6DAF600":
REM Y
9961 DATA "00003C660C663C00":
REM Z
9962 DATA "0000D6D6D6D6FE00":
REM [
9963 DATA "00007C063E067C00":
REM \
9964 DATA "0000D6D6D6D6FF03":
REM ]
9965 DATA "000066663E060600":
REM ^
9966 DATA "00000000000000FF":
REM_
9967 DATA "CEDBDBFBDBDBC
E00": REM`
9968 DATA "0E1E3666FEC6C600":
REM a
9969 DATA "FCC0C0FCC6C6FC
00": REM b
9970 DATA "CCCCCCCCCCCCCF
C06": REM c
9971 DATA "7E6666666666FFC3":
REM d
9972 DATA "FCC0C0F8C0C0FE00":
REM e
9973 DATA "187EDBDBDB7E1800":
REM f
9974 DATA "7C60606060606000":
REM g
9975 DATA "C3663C183C66C300":
REM h
```

> 18

Uwaga: od chwili uruchomienia procedury do momentu przełączenia na nowy zestaw znaków mija ok. 18 sekund.

Procedura oparta jest na pomyśle Wojciecha Zientary.

```
    5 GOSUB 29000
   10 REM * GŁÓWNY
PROGRAM *
   15 END
29000 MEM=PEEK(106)-8:
CHACT=MEM*256
29010 FOR I=0 TO 1023:POKE
CHACT+I,PEEK(57344+I):
NEXT I: RESTORE 30000
29020 READ A:IF A=-1 THEN
POKE 756,MEM:RETURN
29030 FOR I=0 TO 7:READ
B:POKE CHACT+A*8+I,B:
NEXT I: GOTO 29020
30000 DATA 64,0,124,96,124,102,
102,124,0
30001 DATA 65,0,62,107,107,
107,62,8,0
30002 DATA 66,0,102,110,126,
126,118,102,0
30003 DATA 67,0,60,102,96,96,
102,60,0
30004 DATA 68,0,124,102,124,
102,102,124,0
30005 DATA 69,0,102,102,62,6,
102,60,0
30006 DATA 70,0,24,60,102,102,
126,102,0
30007 DATA 71,0,126,102,102,
102,102,102,0
30008 DATA 72,0,124,102,102,
124,96,96,0
30009 DATA 73,0,107,107,107,
107,107,127,0
30010 DATA 74,0,60,102,102,
102,102,60,0
30011 DATA 75,0,30,54,54,54,54,
102,0
30012 DATA 76,0,30,54,54,54,
127,99,0
30013 DATA 77,0,96,96,124,102,
102,124,0
30014 DATA 78,0,126,24,24,24,
24,24,0
30015 DATA 79,0,107,107,107,
107,107,127,7
30016 DATA 80,0,60,102,12,6,
102,60,0
30017 DATA 81,24,102,110,126,
126,118,102,0
30018 DATA 82,0,102,108,120,
120,108,102,0
30019 DATA 83,0,99,123,111,
111,111,123,0
30020 DATA 84,0,126,96,124,96,
96,126,0
30021 DATA 85,0,126,102,96,96,
96,96,0
30022 DATA 86,0,99,119,127,
107,99,99,0
30023 DATA 87,0,102,102,102,
102,102,126,14
30024 DATA 88,0,102,102,102,
62,6,6,0
30025 DATA 89,0,102,102,126,
102,102,102,0
30026 DATA 90,0,62,102,102,62,
54,102,0
30027 DATA 96,0,110,123,123,
123,123,110,0
30028 DATA 123,0,107,107,62,
62,107,107,0
30029 DATA 124,0,112,48,62,51,
51,62,0
30030 DATA 60,0,60,6,62,6,6,60,0
30031 DATA 62,0,102,102,60,60,
102,102,0
30032 DATA -1
```

17 <

```
9976 DATA "C6C6CEDEF6E6C600":
REM i
9977 DATA "D6C6CEDEF6E6C600":
REM j
9978 DATA "C6CCD8F8CCC6C6
00": REM k
9979 DATA "0E1E3666C6C6C60
0": REM l
9980 DATA "C6EEFED6C6C6C600":
REM m
9981 DATA "C6C6C6FEC6C6C600":
REM n
9982 DATA "7CC6C6C6C6C67C00":
REM o
9983 DATA "FEC6C6C6C6C6C600":
REM p
9984 DATA "7EC6C67E3666C600":
REM q
9985 DATA "FCC6C6FCC0C0C000":
REM r
9986 DATA "7CC6C0C0C0C67C00":
REM s
9987 DATA "FC30303030303000":
REM t
9988 DATA "C6C6C67E06C67C00":
REM u
9989 DATA "DBDB7E187EDBDB00":
REM v
9990 DATA "F8CCCCFCC6C6FC00":
REM w
9991 DATA "C0C0C0FCC6C6FC00":
REM x
9992 DATA "C6C6C6F6DADAF600":
REM y
9993 DATA "3C66061C06663C00":
REM z
9994 DATA "D6D6D6D6D6D6FE00":
REM {
9995 DATA "7CC6061E06C67C00":
REM:
9996 DATA "D6D6D6D6D6D6FF03":
REM }
9997 DATA "6666663E06060600":
REM ~
9998 DATA "3C4299A1A199423C":
REM@
9999 REM START 64000,
END 64768
```



Leszek Rudak

Nie lubię rzucać kobietą

Nie jestem antyfeministą. Nie czuję się też brakującym ogniwem w historii człowieka. Z tego wynika, że nie lubię "rzucać" kobietami, a więc nie będę więcej występował na arenach zawodów olimpijskich w prehistorii.

Oczywiście taki początek nie wygląda zniechęcająco ani zachęcająco. Wobec tego potrzebny jest jakiś opis tego, czym usiłowałem się bawić.

Caveman Ugh-lympics jest typową grą w nietypowej otoczce. Gier imitujących różne zawody sportowe na ekranie monitora komputerowego było już sporo. Właściwie na wszystkich komputerach od domowych (np. ZX Spectrum) do profesjonalnych (IBM PC) można znaleźć jakiś stadion lub choćby kawałek bieżni. Opisywana gra jest jeszcze jedną z tej serii. Rozgrywamy zawody w kilku konkurencjach: trzy techniczne, jedna siłowa i dwie biegowe. Nietypowe są natomiast same konkurencje. Na początek wystarczą nazwy: **Mate Toss** – rzut kobietą, **Dino Race** – wyścigi dinozaurów, **Saber Race** – ucieczka przed kłami tygrysa, **Dino Vault** – skok przez dinozaura, **Clubbing** – walka na maczugi, **Fire Making** – rozniecanie ognia. Nietypowy jest też czas zawodów (ok. 75 tys. lat przed naszą erą) i uczestnicy (Ughlympijczycy są neandertalczykami). Niestety reszta jest typowa dla gier typu olimpiady. Cała zabawa polega na jak najszybszym naciskaniu klawiszy i to w większości bez konieczności myślenia. Oczywiście jeżeli ktoś ma joystick, to za pomocą tej gry może poćwiczyć ubijanie kogla-mogla.

Warto może jeszcze bliżej wyjaśnić, na czym polegają konkurencje tej pradawnej olimpiady. Zacznijmy od początku. Mate Toss polega na mocnym uchwyceniu swojej kobiety za włosy (na szczęście wtedy modne były włosy do pasa), rozkręceniu jej wokół siebie i wyrzuceniu daleko, daleko, jak najdalej. Ta konkurencja była pierwowzorem rzutu młotem.

Wyścigi dinozaurów są podobne do wyścigów konnych z przeszkodami. W tej grze ścigamy się z jednym tylko rywalem. Dosiadamy dinozaura (a właściwie coś pośredniego między tyranozaurusem i brontozaurusem), wprawiamy go w ruch i biegniemy do przodu. Kłopoty zaczynają się przy przeszkodach. Nasz Dino nie zawsze ma ochotę przeskoczyć kamień. Cofa się lub skacze za nisko, a wtedy wylatujemy z siodła. W tej konkurencji ekran jest podzielony na dwie części. W dolnej części rywal pogania swojego wierzchowca, a górna część jest cała dla nas i naszego gada. Można przyspieszać i zwalniać bieg dinozaura. Można też pobudzać go do skoku, ale i tak Dino robi co chce.

Idea trzeciej konkurencji zasadza się na starym przysłowiu najstarszych Neandertali: "Wyścigi nie zawsze odbywają się po to, by wskazać najszybszego, ale po to, by nie zostać zjedzonym". W wyścigu bierze udział dwóch zawodników i tygrys. Tygrys biegnie w dolnej połowie ekranu, a praludzie w górnej. Trzeba się spieszyć, bo jeżeli tygrys dobiegnie do prawego brzegu ekranu, to pojawi się tuż za plecami biegaczy i wtedy biada. Oczywiście te parę tysięcy lat temu nie było tartanu, a więc trzeba przeskakiwać rozpadliny i kaktusy. Nie można też liczyć na współpracę rywala – gdy tylko będzie mógł, to utrudni nam bieg (pewnie licząc na to, że tygrys jest na diecie).

Następna konkurencja to zwykły skok o tyczce. Zwykły, oczywiście, w ówczesnych realiach. Mamy do dyspozycji jakiś zwiędły badyl, a musimy przeskoczyć przez wielkiego tyranosurusa. Sterowanie polega na rytmicznym naciskaniu klawiszy tak, by nie pomylić nogi przy rozbiegu, a potem na naciskaniu Enter we właściwych momentach: postawienia tyczki i odbicia się od niej.

Walka na maczugi, piąta konkurencja, składa się z dwóch części. Pierwsza to wzajemne obrzucanie się wyzwiskami i straszenie przeciwnika. Po tym krótkim wstępie zaczyna się prawdziwa walka. Jak zwykle w grach imitujących walkę trzeba kilka razy trafić przeciwnika w głowę lub kolano, a do tego unikać jego ciosów.

Ostatnią konkurencją jest rozniecanie ognia. Autorzy gry dostarczają zapalniczkę składającą się tylko z dwóch patyków. Trzeba jak najszybciej pocierać je o siebie, aż posypią się iskry. Od jednej iskry ognisko nie zapłonie, a więc trzeba jeszcze trochę podmuchać i dopiero wtedy ukaże się wspaniały ogień. W tej konkurencji, jak w poprzednich, nie istnieją zasady fair play. Gdy tylko trochę wyprzedzimy rywala, to on zdzieli nas po głowie swoimi kijkami, co znacznie utrudni nam dmuchanie.

Jeżeli ktoś zafascynuje się tą przedziwną olimpiadą to znajdzie kilka wygodnych elementów w tej grze. Po pierwsze można wybrać jednego z wielu "atletów". Każdy z nich jest specjalistą w jednej z konkurencji, a inne niestety ma gorzej opanowane. Można też trenować wybraną konkurencję tyle razy, ile potrzeba do osiągnięcia mistrzostwa. Po wszechstronnym treningu można przystąpić do zawodów na poważnie. Za każdą konkurencję otrzymuje się punkty i w końcu staje na podium. Można też, co zapewne jest ciekawsze, grać w kilka osób. Jest sześciu atletów, a więc w każdego może wcielić się inny gracz. Rywalizuje się wtedy parami (w wyścigach, walce na maczugi i rozniecaniu ognia) albo po prostu zdobywa jak największą liczbę punktów. Po wszystkich konkurencjach ceremonia dekoracji jest piękną nagrodą za wysiłek.

Dobra grafika jest jednym z atutów tego programu. Niestety jej ceną jest bardzo długie przygotowywanie kolejnej konkurencji, tak długie, że czasem miałem wrażenie, że coś się zapętliło. Trzeba jednak przyznać, że grafika jest dopracowana w szczegółach. Każde niepowodzenie jest odpowiednio i dowcipnie obrazowane. Podobnie dowcipne są animowane filmy z otwarcia olimpiady wraz z zapalaniem znicza i z dekoracji zwycięzców. Odpowiednio dobrana muzyczka i inne dźwięki umilają grę.

Dowcip nie opuszczał autorów także przy charkterystykach zawodników, ale na ekranie teksty wyświetlane są w bardzo nieczytelnej postaci: jako litery niezdarnie wykute w kamieniu.

Do gry dołączona jest instrukcja. Dzięki niej wiadomo o co gramy i co można robić aby grać. W instrukcji są też wskazówki pochodzące od dawnych mistrzów olimpijskich, mówiące ponoć jak wygrywać. Niestety dawni siłacze nie byli zbyt komunikatywni.

Nie mogę wysoko ocenić tej gry. Moim zdaniem jest nudna, mimo ładnej i dowcipnej grafiki.

Gra Caveman Ugh-lympics jest przeznaczona na komputery zgodne z IBM PC, pracujące pod kontrolą systemu MS DOS 2.11 lub wyższym. Wygodniej jest używać komputera wyposażonego w dysk stały, ale można grać mając tylko jeden napęd dysków. Trzeba wtedy mieć 4 dyskietki. Gra ta pracuje z każdą kartą graficzną (nie wymaga emulatora na Herculesie, można skonfigurować na Tandy, itp). Na dysku stałym zajmuje ok 1,3 MB. Z instrukcji wynika, że istnieją wersje tej gry na Commodore C64 i 128.





Oprogramowanie operacji wejścia/wyjścia

Współpraca komputera i urządzenia zewnętrznego sprowadza się do przekazywania danych pomiędzy nimi. Niezależnie bowiem od uwarunkowań sprzętowych (określających m.in. hierarchię urządzeń czy sposób inicjalizacji, prowadzenia i zakończenia wymiany danych) jedynymi programowo dostępnymi operacjami WE/WY są operacje zapisu danych do urządzenia wyjściowego (OUT) oraz operacje wczytania danych z tego urządzenia (IN). Jednak obok czynności wymiany danych, musimy za pomocą tych skromnych środków programowych zrealizować także złożone operacje kontrolno-sterujące. W przypadku pracy z urządzeniami zewnętrznymi magistrala danych służyć musi więc programiście nie tylko jako kanał transmisji, ale także jako magistrala kontrolno-sterująca. Jest to możliwe dzięki bogatym możliwościom sprzętowego przetwarzania informacji — taki sam stan magistrali danych może być odmiennie interpretowany w różnych portach, w zależności od ich rozwiązań układowych.

I tak np. adresując przy operacji wyjścia port do którego dołączony jest przetwornik C/A, musimy jednocześnie podać na magistralę danych wartość binarną potrzebną do realizacji założonej wielkości analogowej. Przesyłamy więc wielkość numeryczną. Jeśli zaś adresujemy port, który steruje pracą przetworników A/C, to stan magistrali danych może decydować o tym, które z przetworników mają rozpocząć przetwarzanie. Informacja z magistrali danych ma więc charakter impulsów sterujących. A jeśli do portu jest przypisany jeden przetwornik, to wyzwalanie przetwornika odbywa się tylko na podstawie dekodowania określonego stanu magistrali adresowej i stan magistrali danych może wtedy być dowolny.

Podobnie przy odczycie danych. Wprowadzana przez magistralę danych do procesora informacja może mieć charakter stricte numeryczny (jak np. odczyt stanu przetwornika A/C) i być bądź bezpośrednio przetwarzana, bądź przesyłana do RAM celem późniejszej analizy. Ale może też być ona interpretowana jako sygnał o określonym stanie urządzenia zewnętrznego, wykorzystywany bezpośrednio do jego programowej obsługi. (Np. przed wczytaniem danych z przetwornika A/C należy sprawdzić czy proces przetwarzania wielkości analogowej na cyfrową został już zakończony. W tym celu możemy odczytywać – do momentu wystąpienia określonego stanu – dane z portu, do którego doprowadzone są te wyjścia przetworników A/C, które informują o zakończeniu przetwarzania).

Należy wspomnieć, że urządzenie zewnętrzne może zasygnalizować procesorowi komputera konieczność zaniechania aktualnie wykonywanego programu i przejścia do procedury obsługi tego urządzenia. Realizuje się to na drodze sprzętowo-programowej z wykorzystaniem przerwań i przy współudziale operacji wejścia/ /wyjścia. Liczne informacje na ten temat można znaleźć w literaturze, a także w "Komputerze" i innych czasopismach komputerowych.

Jak wynika z powyższych przykładów, oprogramowanie urządzeń zewnętrznych musi być ściśle związane z konkretnymi rozwiązaniami sprzętowymi. Z drugiej strony stosunkowo skromny zasób operacji WE/WY procesora uwarunkowuje rozwiązania sprzętowe już na etapie koncepcyjnym. Konieczne wydaje się więc ich omówienie. Efektywne i w czasie rzeczywistym sterowanie pracą urządzeń zewnętrznych jest możliwe jedynie za pomocą języka maszynowego. Musimy z niego korzystać także w PASCAL-u, gdyż standardowo nie posiada on możliwości realizacji operacji WE/ WY. (Aby je zrealizować, należy włączyć do programu instrukcje asemblerowe za pomocą instrukcji kodowej INLINE).

Przesyłanie danych do urządzenia zewnętrznego z poziomu języka maszynowego może odbywać się za pomocą rozkazów: OUT (dwa warianty), OUTI, OUTD, OTIR, OTDR.

Najpierw omówimy rozkaz o mnemoniku:

OUT (C),r

gdzie: C – rejestr C procesora,
r – jeden z rejestrów A,B,C,D,E,H,L,

Powoduje on, że procesor wysyła zawartość rejestru r do portu o adresie określonym zawartością rejestrów C (młodszy bajt adresu) i B (starszy bajt adresu). Inaczej mówiąc, rejestr r określa stan magistrali danych, zaś rejestry BC określają stan magistrali adresowej.

Drugi wariant rozkazu OUT to:

OUT (n),A

gdzie: A – rejestr A procesora,
n – liczba 8 bitowa.

Adres jest tutaj określony wartością n (młodszy bajt) i zawartością A (starszy bajt). Na magistralę danych jest podawana zawartość rejestru A.

Przed użyciem rozkazu OUT (C),r musieliśmy załadować trzy rejestry procesora (B,C i r), zaś teraz tylko jeden (A). Użycie rozkazu OUT (n),A jest więc bardziej optymalne, tak pod względem objętości procedury jak i czasu jej wykonania. Z tego względu rozkaz ten jest korzystniejszy w przypadku, gdy nieistotny jest stan magistrali danych. Jeśli dekodujemy stan magistrali danych, to zastosowanie rozkazu OUT (n),A jest praktycznie żadne i ogranicza się do przypadku, gdy starszy bajt adresu ma identyczną wartość jak wysyłane dane. (Główne zastosowanie tego rozkazu ma miejsce w małych systemach mikroprocesorowych, przy adresowaniu tylko za pomocą młodszego bajtu. W takim przypadku nie dekodujemy starszego bajtu adresu i rejestr A służy tylko do przekazywania danych).

Pozostałe rozkazy z grupy OUT przesyłają zawartość miejsca pamięci adresowanego przez parę rejestrów HL do portu o adresie określonym zawartością rejestrów BC - wykonują operację, którą możemy zapisać jako OUT (BC),(HL). Oprócz tego:

- OUTI – inkrementuje (zwiększa o jeden) wartość rejestru HL i dekrementuje (zmniejsza o jeden) wartość rejestru B,
- OUTD – dekrementuje rejestr HL i B,
- OTIR – powtarza OUTI dopóki B różne od zera,
- OTDR – powtarza OUTD dopóki B różne od zera.

Operacje te są wykorzystywane w celu sekwencyjnego przesyłania danych do większej liczby urządzeń zewnętrznych i są rzadko potrzebne w systemach małej automatyzacji.

Do wczytywania danych z urządzenia zewnętrznego procesor posiada rozkazy: IN (dwa warianty), INI, IND, INIR, INDR. Ich składnia i działanie jest analogiczne jak w przypadku rozkazów z grupy OUT. I tak rozkaz o mnemoniku:

IN r,(C)

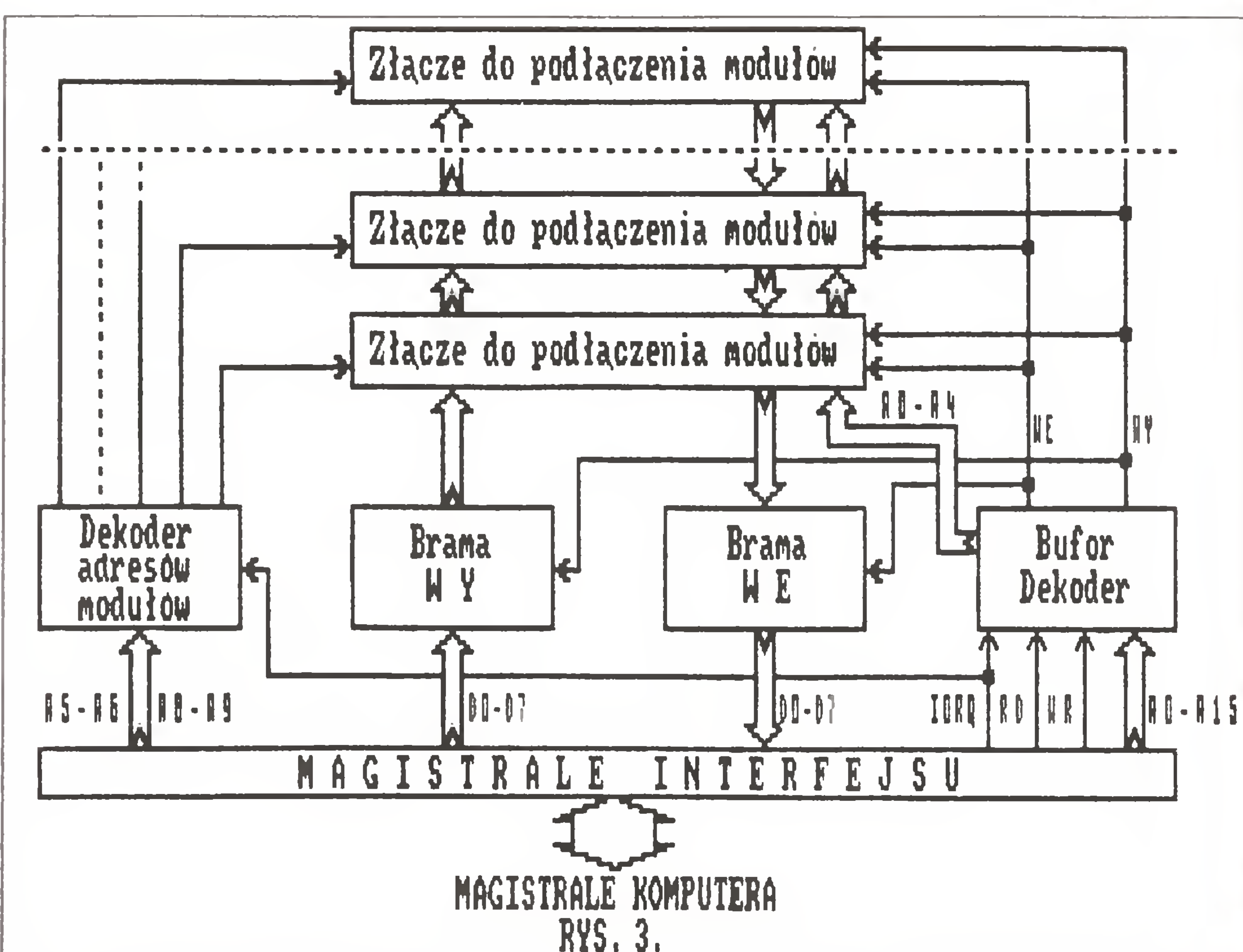
gdzie: C i r jak poprzednio,

wczytuje do rejestru r procesora zawartość portu o adresie określonym zawartością rejestrów C (młodszy bajt adresu) i B (starszy bajt adresu). W rejestrze r znajdzie się więc zawartość portu określonego przez zawartość rejestru BC. Drugi wariant rozkazu IN to:

IN A,(n)

gdzie: A i n jak poprzednio.

> 20



RYS. 3.

19 <

Adres jest tutaj określony wartością n (młodszy bajt) i zawartością A (starszy bajt). Zawartość magistrali danych jest wczytywana do akumulatora (rejestr A procesora). W tym przypadku wprowadzona informacja zniszczy starszy bajt adresu. Rozkaz ten, podobnie jak OUT (n),A jest także bardziej "ekonomiczny" od swego poprzednika. Lecz teraz tylko pod warunkiem, że nie trzeba będzie go powtarzać. (Chcąc powtórzyć operację, będziemy musieli powtórnie ładować akumulator starszym bajtem adresu).

Pozostałe rozkazy z grupy IN przesyłają zawartość portu o adresie określonym zawartością rejestrów BC do miejsca pamięci o adresie określonym przez zawartość rejestrów HL – wykonują operację IN (HL),(BC).Ich mnemoniki i działanie są następujące:

- INI – inkrementuje wartość rejestru HL i dekrementuje wartość rejestru B
- IND – dekrementuje rejestr HL i B,
- INIR – powtarza INI dopóki B różne od zera,
- INDR – powtarza IND dopóki B różne od zera.

Oczywiście przy oprogramowywaniu urządzeń zewnętrznych, oprócz omówionych powyżej rozkazów z grupy OUT i IN używać musimy także wielu innych – choćby do ładowania rejestrów, zarządzania pamięcią czy przeprowadzania obliczeń. Struktura programu sterującego zależeć będzie od zastosowanych rozwiązań sprzętowych oraz od charakteru realizowanych przez ten sprzęt czynności. Programowanie odbywa się na ogólnie znanych zasadach, a różnorodność możliwych rozwiązań sprawia, że trudno jest udzielić konkretnych wskazówek. Należy jedynie pamiętać, że sterując procesem w czasie rzeczywistym musimy zapewnić odpowiednią charakterystykę czasową algorytmu. Pojedyncza operacja realizowana w języku maszynowym wykonuje się kilka mikrosekund (4-23 taktów zegarowych), a potrzebne dane na ten temat można znaleźć w literaturze.

Możliwe jest także sterowanie urządzeniami zewnętrznymi za pomocą Basic-a. Czasy realizacji komend Basicowych są odpowiednio dłuższe i wynoszą przeważnie kilka milisekund. W przypadku niektórych komend czas ich realizacji może zależeć od rodzaju i wartości użytego argumentu. W razie potrzeby można go wyznaczyć za pomocą programu testującego. (W jednej linii programowej umieszczamy pustą pętlę FOR-NEXT o dużej – np. 10 tys. -liczbie powtórzeń i mierzymy czas jej wykonania. Następnie do środka pętli wstawiamy badaną operację i powtórnie mierzymy czas wykonania. Różnica czasów podzielona przez ilość obiegów pętli da nam szukaną wartość).

Najczęściej Basic (lub Pascal) łączymy z procedurami w języku maszynowym, realizując przy ich pomocy newralgiczne czasowo operacje. Pozwala to zoptymalizować czas opracowywania i uruchamiania oprogramowania, z jednoczesnym zachowaniem możliwości stosunkowo łatwej i szybkiej modyfikacji algorytmu. Sterowanie urządzeń zewnętrznych przy pomocy programu napisanego wyłącznie w BASIC-u jest możliwe jedynie w przypadku stosunkowo wolnych i mało złożonych układów automatyki. Przy omawianiu BASIC-owych operacji WE/WY instrukcja obsługi komputera ogranicza się do podania ich składni oraz zaznaczenia, że powinny być one używane jedynie przez doświadczonego programistę. Ponieważ "nie taki diabeł straszny jak go malują", więc poświęćmy im nieco więcej miejsca niż autorzy instrukcji.

Wysyłanie danych do urządzenia zewnętrznego z poziomu BASIC-a realizuje rozkaz OUT /1/. Jego składnia jest następująca:

OUT m,n

gdzie: m – adres portu (liczba naturalna z zakresu 0 – 65535),
n – wartość bajtu podawanego na szynę danych (liczba naturalna z zakresu 0 – 255).

Interpreter wykorzystuje do realizacji tej komendy rozkaz asamblerowy OUT (C),r ładując do rejestrów BC wartość m, zaś do rejestru r wartość n. Warto tu przypomnieć, że w Basicu Amstrada można podawać liczby zarówno w zapisie heksadecymalnym (z przedrostkiem "&") jak i w zapisie binarnym (z przedrostkiem "&X"). Do portów nie używanych przez system możemy bez obawy wysyłać dowolne wartości danych. Oczywiście pod warunkiem, że nie dołączyliśmy jeszcze "swoich" urządzeń zewnętrznych. Jeśli zaś je dołączymy, chyba będziemy wiedzieć co, gdzie i z jakim skutkiem możemy wysłać.

Wczytanie danych do komputera umożliwia funkcja INP (INput Port):

INP(m)

gdzie: m – jak poprzednio.

W wyniku działania tej funkcji otrzymujemy zawartość portu o adresie m. Oczywiście funkcję INP musimy użyć łącznie z rozkazem lokującym zwracaną przez tą funkcję z portu wejściowego wartość. Na przykład: odczyt=INP(adresportu). Albo: PRINT INP (F89C). Czy też: POKE bufor,INP(adresportu). Dane do zmiennych i tablic wczytywać możemy zupełnie bezkarnie, ale do pamięci RAM (za pomocą rozkazu POKE) tylko "pomiędzy" BASIC i zmienne systemowe. Jako bufor do przechowywania wczytanych danych może być więc użyta pamięć RAM od adresu o jeden większego od wartości ustalonej przez nas rozkazem MEMORY, do adresu 42747 (A6FB hex) włącznie. (Zakładając, że przed MEMORY użyliśmy SYMBOL AFTER 256, likwidując w ten sposób obszar zarezerwowany na znaki definiowane przez użytkownika). Możliwe jest też zapamiętanie wczytanych danych w drugim, nie używanym przez BASIC banku RAM. ("Komputer" publikował już informacje na ten temat w nr. 5/90).

Basic Amstrada CPC 6128 dysponuje również bardzo użytecznym rozkazem WAIT, który umożliwia testowanie stanu portu. Rozkaz ten ma następującą składnię:

WAIT m,a,b

gdzie: m – adres testowanego portu (jak wyżej),
a – wartość maski dla operacji logicznej "i" (AND),
b – wartość maski dla operacji logicznej "wyłącznie-lub" (EXCLUSIVE-OR).

Argumenty a i b powinny być liczbami naturalnymi z zakresu 0-255. Argument b jest opcjonalny i możemy go nie używać (pomijamy wtedy również poprzedzający go przecinek). Rozkaz ten zatrzymuje działanie programu do czasu odczytania z zadanego portu określonej wartości. Przy czym polega to na tym, że najpierw na odczytywanej z portu wartości oraz na argumencie b jest wykonywana operacja logiczna EXCLUSIVE-OR. (Jeśli argument b pominęliśmy, to operacja ta oczywiście nie będzie realizowana). Następnie na wyniku tego porównania i na argumencie a jest wykonywana operacja logiczna AND. Program jest kontynuowany dopiero wtedy, gdy wynikiem finalnym jest liczba różna od zera. Wartości argumentów a i b są przy tym traktowane binarnie i porównywane bit po bicie ze stanem magistrali danych.

Ten nieco zawiły opis nabierze jasności po przeanalizowaniu poniższego przykładu. Wprowadźmy dodatkowe oznaczenia:

p – wartość odczytana z portu o adresie m podczas realizacji rozkazu WAIT,
e – wynik operacji logicznej EXCLUSIVE-OR
w – wynik końcowy operacji logicznych realizowanych

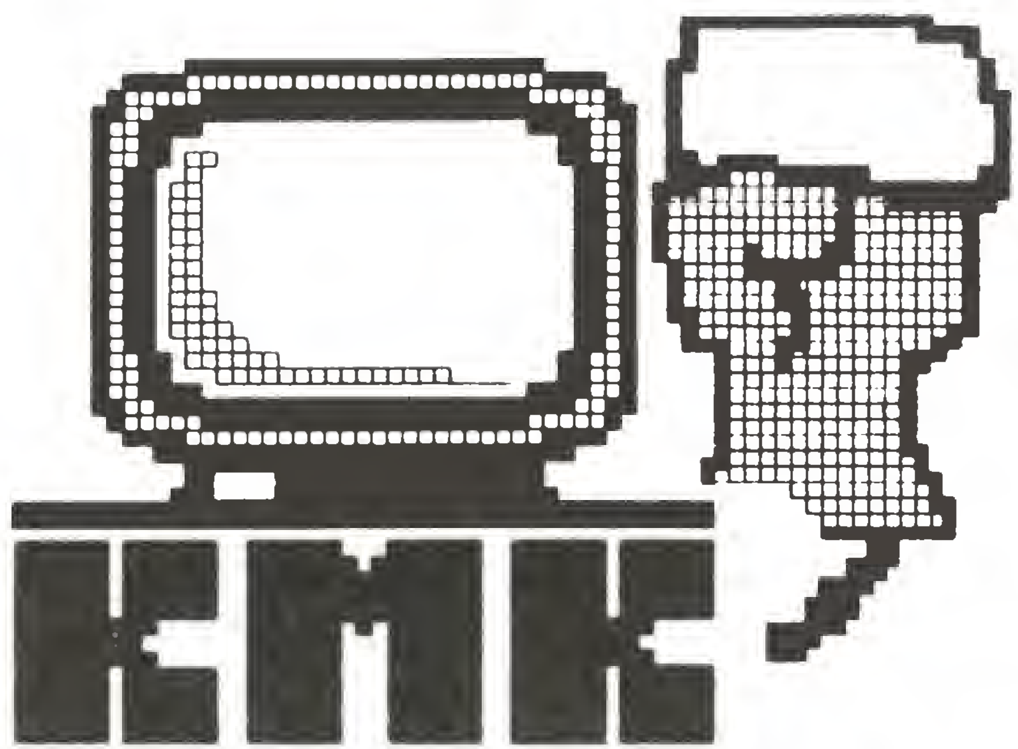
Jeśli zadamy np. a=240, b=85, zaś komputer odczyta z portu o adresie m przykładowo wartość p=15, to realizacja rozkazu WAIT m,240,85 będzie przebiegać następująco:

```
========
76543210   – numery bitów (linii magistrali danych)
========
00001111   – binarna postać wartości odczytu portu p=15
01010101   – binarna postać maski b=85
--------
01011010   – binarny wynik operacji: p EXCLUSIVE-OR b = e
11110000   – binarna postać maski a=240
--------
01010000   – binarny wynik operacji: e AND a = w
```

Decymalnie w=80 i ponieważ jest to wartość różna od zera, więc po odczytaniu z portu wartości p=15 interpreter przejdzie do wykonywania kolejnego rozkazu Basica. Możemy sprawdzić, że wynik w=0 otrzymamy tylko wtedy, gdy p przyjmuje takie wartości, iż bity 4 i 6 mają wartość "1", zaś bity 5 i 7 wartość "0". (Stan czterech najmłodszych bitów nie ma tu wpływu na wynik końcowy). Sytuacja taka występuje gdy wartości wczytywane z testowanego portu mieszczą się w zakresie od p=80 do p=95 włącznie. W takim przypadku komputer nie przejdzie do wykonywania następnego rozkazu, aż do chwili zmiany stanu któregokolwiek ze starszych bitów na przeciwny. Tak więc dobierając odpowiednie wartości argumentów a i b możemy sprawić, że kontynuacja lub wstrzymanie działania programu zależeć będzie od określonego stanu magistrali danych. Należy jedynie pamiętać, iż jeśli sprzęt nie dostarczy do odczytu odpowiedniej wartości, to przerwać testowanie portu można jedynie przez wykonanie operacji RESET lub wyłączenie zasilania komputera.

Pozostałe operacje Basica są wystarczająco dokładnie opisane w instrukcji i nie ma potrzeby ich tutaj omawiać. Można jedynie zachęcić do jej uważnego studiowania, gdyż Basic 1.1 opracowany przez Locomotiv Software Ltd oferuje właścicielowi Amstrada CPC 6128 możliwości niespotykane w innych komputerach tej klasy. Oprócz bardzo bogatego zestawu funkcji arytmetycznych, tekstowych i graficznych mamy możliwość deklarowania typów zmiennych, dysponujemy istrukcjami GOTO i GOSUB z selektorami skoku oraz pętlami IF-

-THEN-ELSE i WHILE-WEND. Możemy też przedefiniować klawiaturę, zdefiniować własne znaki, a do 32 klawiszy przypisać całe ciągi znaków. Mamy możliwość wywoływania podprogramów po określonym czasie i co określony czas (przerwania programowe!). Możemy przejąć obsługę błędów systemowych i zdefiniować 8 okien. Przy takim bogactwie środków programowych efektywne oprogramowanie urządzeń zewnętrznych nie powinno nastręczać problemów.



PĘTLICZEK – bo pętla jest podstawą programowania. Tu znajdziesz kolejną porcję zadań naszego Klubu Mistrzów Komputera.
MĘTLICZEK – bo znajdziesz tu różne różności, związane z minikomputerem tak cienką nitką, że Redakcja już nie bierze za nią odpowiedzialności.
Redakcja strony klubowej: Marcin Jędrzejewski, Leszek Rudak.

ZADANIA KLUBOWE

7/90. Proszę opracować program składający kwadratową kartkę papieru. Po podaniu ilości złożeń program powinien wyrysować tę kartkę z zaznaczonymi liniami zagięć. Linie powinny uwzględniać kierunek zagięcia, aby odzwierciedlić charakterystyczne wybrzuszenia rzeczywistej kartki poddanej serii zagięć i następnie rozwiniętej.

(zadanie nadesłał Tadeusz Binek z Krakowa)

8/90. W książce "Opowieści matematyczne" dr Michał Szurek podaje jak wyrazić 1983 za pomocą jedynek: $1983 = (1+1+1)*\{1+(1+1+1)*(1+1+1+1)*(1+1+1+1+1)*[(1+1+1)*(1+1+1)+1+1]\}$

Proponuję napisać program podający dla danej liczby minimalną liczbę jedynek pozwalającą wyrazić tę liczbę (dopuszczalnymi operacjami są tylko dodawanie i mnożenie; nie wolno łączyć jedynek w liczby wielocyfrowe).

L.R.

9/90. Proszę napisać prosty program rezydujący w pamięci komputera, który wyświetla zawartość rejestrów wewnętrznych procesora.

(zadanie nadesłał Jerzy Labocha ze Szczecina)

FILM

Pod koniec XX wieku wszyscy już wiemy jak powstaje film. Cała tajemnica ukryta jest w tysiącach klatek pokazywanych bardzo szybko jedna za drugą. Klatki troszeczkę różnią się między sobą i dzięki temu na ekranie mamy ruch. Filmy animowane powstają w taki sam sposób, ale wszystkie klatki trzeba narysować.

Gdy zamierzamy zrobić animację na ekranie monitora komputerowego to musimy wybrać sposób jej wykonania. W przybliżeniu można powiedzieć, że mamy do wyboru dwie metody animacji. Jedna to jakoś wprowadzić do pamięci komputera gotowe rysunki i pokazywać je bardzo szybko po kolei. Druga, to zapisać rysunek w postaci formuł obliczeniowych i wyświetlać wyniki obliczeń dla argumentów zmieniających się w odpowiednim tempie. Obie te metody mają wady. Najlepszy wybór to rozwiązanie mieszane: trochę obliczeń i trochę gotowych wzorów. Poniżej przedstawiam bardzo prosty program wykorzystujący pierwszą metodę animacji. Program ten oczywiście nie kandyduje do "Oscara", ale może służyć jako zabawka zamiast słynnego kowala nawijanego na ołówek.

Program napisany jest w dialekcie Turbo Pascal 5.0, ale oczywiście można go skompilować za pomocą dowolnego kompilatora po dostosowaniu kilku drobiazgów (poprzedzonych na wydruku komentarzem {TP 5.0}). Myślę, że Mistrzom nie muszę wyjaśniać działania programu.

```
program anima;
{TP 5.0} uses crt;
{TP 5.0} const MAX=80;
var i,j,k:integer;
    wolne:string;
    rysunek:array [0..3,1..5] of
    string;
begin
{TP 5.0} Clrscr; {czyszczenie
ekranu}
 for k:=0 to 3 do
 begin
    writeln('Podaj rysunek (5x5)
nr: ',k+1);
    for i:=1 to 5 do readln(rysu-
nek[k,i]);
    writeln
 end;
{TP 5.0} Clrscr;
 wolne:='';
 for i:=1 to MAX-5 do
 begin
   for j:=1 to 5 do
      writeln(wolne,rysunek[i
mod 4,j]);
{TP 5.0} Delay(85); {opóźnienie
85 milisekund}
{TP 5.0} Clrscr;
   wolne:=wolne+' '
 end
end.
```

Do testowania programu wykorzystywałem następujące podstawienia (zamiast podawania za każdym razem rysunku z klawiatury):

```
rysunek[0,1]:=' O ';
rysunek[1,1]:=' O ';
rysunek[0,2]:='/#\ ';
rysunek[1,2]:='/#\ ';
rysunek[0,3]:=' :#';
rysunek[1,3]:=' \ # \';
rysunek[0,4]:='\ ';
rysunek[1,4]:='';
rysunek[0,5]:='';
rysunek[1,5]:=' \';
rysunek[2,1]:=' O ';
rysunek[3,1]:=' O ';
rysunek[2,2]:=' #\ ';
rysunek[3,2]:='/#\ ';
rysunek[2,3]:=' #\ \';
rysunek[3,3]:='/ #/ ';
rysunek[2,4]:='/ \ ';
rysunek[3,4]:='';
rysunek[2,5]:='/ / ';
rysunek[3,5]:='/ ';
```

L.R.

Już od dłuższego czasu Pascal jest "urzędowym" językiem Klubu. W języku tym są dogodne warunki do definiowania podprogramów – procedur i funkcji, dzięki którym uzyskujemy przejrzystą, blokową strukturę programu. Procedura lub funkcja może samodzielnie realizować różne zadania, łatwo też ją wpleść w tekst złożonego programu. Zapewne każdy z nas ma swoją bibliotekę uniwersalnych procedur realizujących pewne często spotykane problemy. Pisząc program dołącza się tylko gotowe i przetestowane podprogramy. Proponuję, aby na stronie klubowej były prezentowane różne pożyteczne podprogramiki, które będziemy mogli umieścić potem w naszych bibliotekach. Zachęcam wszystkich do nadsyłania swoich propozycji. Oczywiście będą one uwzględniane w klubowej punktacji. Zaznaczam, że propozycje te powinny mieć możliwie uniwersalny charakter – standardowe obliczenia, generowanie rozmaitych raportów, operacje na plikach, tablicach itp.

Jak nazwać nasz nowy dział – może tak:

KĄCIK UŻYTECZNYCH PODPROGRAMÓW

Dziś zajmiemy się ponownie permutacjami. Przypomnę, że permutacją zbioru n elementowego X nazywamy każdą funkcję różnowartościową f : X -> X. Permutacje dzielimy na parzyste i nieparzyste. Tematem zadania klubowego 8.89 było określenie parzystości permutacji. Ciekawą propozycję rozwiązania nadesłał **Tomasz Płonka** z Tarnowskich Gór. Aby uprościć rozwiązanie Czytelnik wprowadził pojęcie inwersji dla permutacji zbioru $X = \{1, 2, 3,..., n\}$. Inwersją w permutacji (k(1), k(2), ..., k(n)) nazywamy każdą parę (k(i), k(j)) taką, że $i < j$ oraz $k(i) > k(j)$. Permutację tę nazywamy parzystą, jeżeli ilość inwersji w tej permutacji jest parzysta lub równa zero.

Prześledźmy tę metodę na prostym przykładzie: zbadamy, czy permutacja (2, 5, 1, 4, 3) jest parzysta. Wypiszemy wszystkie inwersje tej permutacji. Jedyną inwersją, w której na pierwszym miejscu występuje 2 jest (2, 1). Dla 5 mamy 3 inwersje: (5, 1), (5, 4), (5, 3). Liczba 1 nie może stać na pierwszym w żadnej inwersji. Pozostaje jeszcze (4, 3) – razem jest ich 5 i permutacja jest nieparzysta.

Wobec powyższych uwag napisanie stosownej funkcji badającej parzystość permutacji jest proste. Poniżej zamieszczam nadesłaną propozycję:

```
function parzysta( a : tablica ) :
boolean;
  var i, j, licznik, p : integer;
  begin
    licznik := 0;
    for i := 1 to MAX-1 do
      begin
        p := a[i];
        for j := i + 1 to MAX do
          if a[j] < p then
            licznik := licznik + 1
      end;
    if odd(licznik) then
      parzysta := false
    else
      parzysta := true
  end;
```

Argumentem funkcji jest tablica liczb typu integer, do której wpiszemy permutację. Skorzystamy więc z deklaracji type tablica = array[1..MAX] of integer;

Wiadomo, że rozmiar tablicy musi być określony w jej deklaracji – stąd konieczność uprzedniego zadeklarowania stałej MAX o odpowiedniej wartości. Podprogram działa zgodnie z omówioną wcześniej zasadą: dla kolejnych liczb bada istnienie ewentualnych inwersji i zlicza je, a na końcu nadaje funkcji odpowiednią wartość logiczną (true dla permutacji parzystej, false dla nieparzystej). Standardowa funkcja Pascala odd() daje w wyniku true, jeżeli jej argument jest liczbą nieparzystą (dla parzystej – false). Zapis można uprościć do: parzysta := not(odd(licznik))

Powyższy przykład dowodzi, że warto pokusić się o pełną analizę problemu, gdyż można w ten sposób otrzymać proste rozwiązanie.

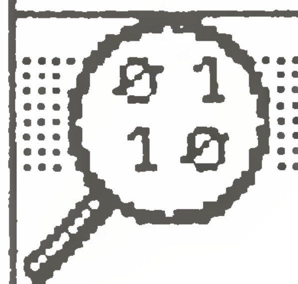
M.J.

PRAWO PIERCE'A

Podczas pierwszego uruchomienia w każdym systemie komputerowym maszyna będzie błędnie interpretować polecenia, mylić się w druku, źle liczyć wyrażenia matematyczne lub przynajmniej ignorować urządzenia zewnętrzne.



Jarosław Kania



Turbo Prolog wersja 2.0

Omawianie konkretnej implementacji języka programowania jest dość niewdzięcznym zajęciem, zwłaszcza gdy istnieją uzasadnione podstawy do przypuszczeń, że duża grupa Czytelników nie zetknęła się z żadną z jego wersji. Głównym problemem jest właściwe wyważenie proporcji między opisem samego języka a opisem jego implementacji – dla Czytelników znających już ten język jego opis będzie nudny, a dla reszty rozważania o konkretnych rozwiązaniach mogą być niezrozumiałe. Mimo wszystko podjąłem się jednak opisu języka Turbo Prolog (wersja 2.0), a uczyniłem to z dwóch powodów. Po pierwsze, jego autorem jest firma Borland International, powszechnie znana z opracowań innych języków programowania, które de facto stawały się punktem odniesienia dla pozostałych implementacji (sztandarowym przykładem jest tu Turbo Pascal). Po drugie, chodzi tu o Prolog, język wciąż jeszcze nie doceniany przez większość programistów, a z pewnością wart szerszego rozpowszechnienia.

Prolog

Prolog (nazwa powstała od słów PROgramming in LOGic - programowanie w logice) pojawił się oficjalnie na początku lat siedemdziesiątych. Zainteresowanie nim wzrosło po zapowiedzi, że ma być podstawowym językiem w tzw. komputerach piątej generacji. Od razu przy pierwszym kontakcie z tym językiem dostrzegamy jego odmienność od tych, do których przywykliśmy, jak np. Pascal, C, Basic itp., czyli tzw. językami imperatywnymi (zwanymi tak dlatego, że wprost określają przebieg sterowania programu – opisują, jak rozwiązać konkretny problem). W przeciwieństwie do nich Prolog, a generalnie wszystkie języki deklaratywne, których Prolog jest typowym przedstawicielem, opisują, co ma być zrobione, bez dokładnego precyzowania sposobów osiągnięcia rozwiązania.

Program napisany w Prologu można traktować jako zbiór faktów i reguł, pozwalających z jednych faktów wydedukować inne. Wszystkie reguły mają postać "jeśli zachodzą fakty F1,...,Fk, to zachodzi fakt F". Zapisujemy to w następujący sposób:

F:-F1.F2,....Fk

Możemy również wyrazić alternatywę pewnych faktów, np. zapis
F:-F1,F2,...,Fk
F:-G1,G2,...,Gn
odczytujemy następująco: "jeśli zachodzą fakty F1 i F2 i... i Fk lub zachodzą fakty G1 i G2 i ... i Gn, to zachodzi fakt F".

Czym jest w języku Prolog fakt? W dużym przybliżeniu fakt odpowiada relacji zachodzącej między pewnymi obiektami, np. informację, że Jan mieszka w Warszawie zapisujemy w Prologu następująco:

mieszka(jan, warszawa)
Jest to ogólna postać zapisu relacji w języku Prolog – nazwa relacji (zależności), a po niej w nawiasach lista obiektów, których ta relacja dotyczy. Relacje mogą być: wieloargumentowe, jak wyżej, jednoargumentowe, np. rudy(jan) a nawet zeroargumentowe, np. w porządku (ten ostatni fakt oznacza po prostu, że jest w porządku).

W Prologu oznaczenie relacji między obiektami nazywamy predykatem n-argumentowym (n=0,1,...). Zamiast pojęcia "reguła" przyjęło się określenie "klauzula". Nazewnictwo to zaczerpnięte jest z logiki matematycznej, z której przecież Prolog się wywodzi.

O ile fakty muszą zawierać konkretne informacje (argumenty predykatów muszą być stałymi), o tyle predykaty w klauzulach mogą zawierać zmienne. W Turbo Prologu (i wielu innych implementacjach) zmienne muszą rozpoczynać się od dużej litery, dlatego w powyższych przykładach "jan" i "warszawa" pisane były małą literą. Przykładowa reguła zawierająca zmienne może wyglądać następująco:

ssak(X):-człowiek(X).
(każdy człowiek jest ssakiem lub, inaczej, jeśli X jest człowiekiem, to X jest ssakiem). Gdy w programie mamy informację, że Jan jest człowiekiem:

człowiek(jan).
to możemy wydedukować, że Jan jest ssakiem. Jak taka dedukcja się odbywa? Mając do wykazania fakt "ssak(jan)" Prolog sprawdza po kolei wszystkie reguły (ich nagłówki, czyli pierwszy człon) i fakty postaci "ssak(...)", próbując dokonać uzgodnienia parametrów. Uzgadnialne są: stała ze stałą, jeśli obie stałe są jednakowe, stała ze zmienną (odpowiednik nadania zmiennej uzgodnionej wartości; w Prologu lepszym określeniem, bardziej oddającym istotę tego procesu, będzie związanie konkretnej wartości ze zmienną) oraz zmienna ze zmienną. W tym ostatnim przypadku związanie pewnej wartości z którąkolwiek z tych zmiennych automatycznie związuje tę wartość z drugą zmienną. W naszym przykładzie następuje uzgodnienie zmiennej X ze stałą "jan" i próba udowodnienia faktu "człowiek(jan)". Ponieważ fakt ten jest dany wprost, tezę "ssak(jan)" uznajemy za udowodnioną.

Przy tym przykładzie zwróćmy jeszcze uwagę na wielofunkcyjność reguł Prologu. Z powyższego zestawu reguł możemy nie tylko wydedukować, że Jan rzeczywiście jest ssakiem, ale także znaleźć odpowiedź na pytanie, kto jest ssakiem, zadając pytanie

ssak(Kto)
Ten sam opisany wyżej mechanizm dokona uzgodnienia i związania zmiennej "Kto" ze stałą "jan". W Prologu ta sama reguła może nam dać różne informacje, w zależności od tego, które parametry są stałymi, a które zmiennymi. Jeśli nie powiedzie się udowodnienie wszystkich faktów z treści reguły, następuje anulowanie związań dokonanych od momentu wybrania konkretnej reguły i podejmowana jest próba wybrania innego jej wariantu, jeśli taki istnieje. Pamiętajmy, że może istnieć kilka reguł dotyczących tego samego predykatu (tej samej relacji).

Na marginesie zauważmy jeszcze jedną charakterystyczną cechę programów w Prologu: brak zmiennych globalnych. Jedyne zmienne, jakie mogą się pojawiać, to te w parametrach klauzul, a związanie z nimi konkretnej wartości trwa do momentu wycofania się do miejsca w programie, w którym nastąpiło uzgodnienie.

Na reguły w Prologu możemy patrzeć także w inny, proceduralny sposób. Mając regułę F:-F1,F2. możemy zamiast interpretacji "F jest prawdą, jeśli prawdą jest F1 i F2" przyjąć interpretację następującą: "wykonanie procedury F polega na wykonaniu instrukcji F1 i F2". Ta interpretacja jest bliższa intuicji zwłaszcza w przypadku predykatów standardowych, gdzie tak naprawdę interesuje nas efekt uboczny związany z danym predykatem. Typowym przykładem jest tu predykat write, którego wykonanie zawsze się powiedzie ("jest zawsze prawdziwy"), ale stosujemy go nie po to, aby się o tym przekonać, lecz aby spowodować wypisanie pewnej informacji. W tym ujęciu na regułę patrzymy jako na pewien sposób nadania wartości niektórym zmiennym oraz spowodowania pewnych efektów ubocznych (wypisania komunikatu, otwarcia pliku itp.).



Istnienie efektów ubocznych i pewne mechanizmy sterowania (odcięcie – cut) są tym, co różni Prolog od czystego programowania w logice, a jednocześnie umożliwia traktowanie go jako języka programowania. I to języka bardzo sprawnego – tekst programu w Prologu jest średnio dziesięciokrotnie mniejszy od tekstu równoważnego mu programu w Pascalu! Biorąc pod uwagę, że badania nad wydajnością programistów wykazują, iż średnia dzienna wydajność mierzona w liniach kodu jest podobna niezależnie od używanego języka (zgadnij, Czytelniku, ile też może wynosić ta wydajność przy pracy nad dużym systemem? Dla ułatwienia dodam, że wystarczy Ci do tego liczba palców u rąk i nóg ...) widzimy, dlaczego Prolog jest językiem wartym poznania.

Ten bardzo skrócony opis nie miał za zadanie zapoznania Czytelnika z językiem Prolog – jego głównym celem było dostarczenie pewnych uwag co do charakteru tego języka, co przy braku miejsca musiało prowadzić do wielu uproszczeń.

Turbo Prolog

Przyjrzyjmy się teraz bliżej konkretnej implementacji Prologu, jaką jest Turbo Prolog (wersja 2.0) firmy Borland International.

Najpierw trochę danych ogólnych: nabywając pakiet Turbo Prologu użytkownik otrzymuje 4 dyskietki i 2 obszerne tomy dokumentacji: podręcznik użytkownika (User's Guide) i kompendium (Reference Guide). Nie zakłada się uprzedniej znajomości jakiejkolwiek wersji Prologu przez użytkownika, choć niewątpliwie ułatwia ona pierwszy kontakt z językiem.

Pierwszy tom dokumentacji (podręcznik użytkownika) zawiera wiele różnorodnych przykładów, dość dokładnie ilustrujących różne aspekty języka. Wszystkie te przykłady znajdują się również na dyskietce. W kompendium, wolałbym, aby standardowe predykaty pogrupowane były według celów, do których służą (np. oddzielnie wejście – wyjście, oddzielnie bazy danych itp.), choć zdaję sobie sprawę, że może to być dyskusyjne. Poza tym informacje podane w tej części są wyczerpujące (może poza jednym wyjątkiem – nie znalazłem informacji, czy i jak można spowodować, aby wykonanie procedury napisanej w innym języku i dołączonej do Turbo Prologu "zawiodło" powodując nawrót).

Po zainstalowaniu pakietu i uruchomieniu głównego programu znajdujemy się w środowisku znanym już z Turbo Pascala (wersja 4.0 i wyżej) oraz Turbo C. Operacje na plikach, wbudowany edytor, zmiana wielkości okien itp. są równie wygodne jak we wspomnianych pakietach. W dostarczanym przez producenta pakiecie bra-

> 24

KOMPUTER test

23 <

kowało mi samodzielnego kompilatora – kompilację można wykonywać jedynie ze zintegrowanego środowiska, co nie zawsze jest wygodne i może nastręczać pewne problemy (pamięć!) przy dużych programach. Mniej wygodne jest także, w porównaniu z innymi produktami firmy Borland, korzystanie z podpowiedzi programu (help), dostępnych jedynie w edytorze.

Turbo Prolog jest kompilatorem, a nie interpreterem jak większość realizacji Prologu. Brak możliwości pracy w trybie interpretacyjnym stanowi pewną wadę, choć nie tak wielką – po skompilowaniu programu, który ma pustą sekcję dyrektyw (o sekcjach piszę niżej) użytkownik pytany jest o dyrektywę, którą może być jedna ze zdefiniowanych w programie klauzul (z dowolnymi parametrami) lub jeden ze standardowych predykatów Turbo Prologu. Wraz z bogatymi możliwościami śledzenia (trace) stwarza to możliwości równoważne pracy interakcyjnej.

Często można spotkać się z poglądem, że tak naprawdę Turbo Prolog nie jest jedynie jedną z możliwych implementacji Prologu, ale stanowi jakby osobny język. Głównym powodem takich stwierdzeń jest podstawowa różnica między Prologiem "klasycznym" a Turbo Prologiem – istnienie typów. W Turbo Prologu każda relacja musi mieć określone typy argumentów. Nie można zatem napisać jednej procedury do odwracania listy – musimy mieć oddzielną procedurę do odwracania listy liczb całkowitych, osobną do odwracania listy napisów itd. (można co prawda trochę "oszukiwać", definiując oddzielny typ jako sumę kilku innych typów, ale istota problemu pozostaje bez zmian). Oprócz wad podejście to ma również swoje zalety:

- znacznie ułatwiony proces kompilacji
- szybszy i mniejszy kod wynikowy
- możliwość wykrycia wielu błędów logicznych w programie na etapie kompilacji (bardzo pomocne!)
- ułatwienie lub wręcz umożliwienie łączenia programów w Prologu z innymi językami.

Kolejną zasadniczą różnicą jest brak mechanizmów meta- programowania, tzn. dynamicznego operowania na programie w trakcie jego wykonywania. Chodzi tu o dodawanie, usuwanie i zmianę reguł w czasie działania programu, czyli o jego automodyfikację. O ile np. w assemblerze automodyfikacja kodu uważana jest słusznie za niezbyt elegancką praktykę i źródło wielu potencjalnych, trudnych do wykrycia błędów, to w Prologu ten mechanizm jest często wykorzystywany i zwiększa siłę języka. Jednym z obszarów zastosowań są np. systemy ekspertowe (expert systems), gdzie dodawanie nowych reguł wnioskowania na podstawie nowych faktów jest bardzo pożądane. Konsekwencją ograniczenia operowania na regułach i faktach jako danych jest brak możliwości przekazywania predykatów i reguł jako parametrów (istnieją tu pewne analogie z przekazywaniem procedur jako parametrów w językach typu Pascal).

Trzecią, może już mniej istotną różnicą, jest brak możliwości definiowania nowych operatorów (ich składni i priorytetów), do czego jest przyzwyczajonych wiele osób pracujących z klasycznym Prologiem. Ten ostatni umożliwia np. zdefiniowanie prawostronnie łącznego operatora '&' (logiczne 'AND') i pisanie a&b&c zamiast AND(a,b,c). Daje się także odczuć brak bardzo silnego narzędzia, jakim są w Prologu gramatyki metamorficzne.

Program w Turbo Prologu podzielony jest na kilka sekcji (to również jest różnica w porównaniu ze standardowym Prologiem).

● Sekcja definicji typów (domain section)

Turbo Prolog ma zdefiniowanych kilka typów pierwotnych, z których podstawowe to integer, real, char, string i symbol. Użytkownik może definiować nowe typy; najczęściej używane metody konstrukcji nowych typów z już zdefiniowanych to alternatywa, np.

```
mójtyp = typ1 ; typ2
```

definiowanie listy obiektów danego typu, np.




```
listaliczb = integer *
```

oraz konstrukcja typów złożonych w sposób przypominający definiowanie rekordów w Pascalu:

```
osoba = opis_osoby(string, integer)
```

Można również definiować własne typy oparte na tej samej dziedzinie, ale traktowane przez Turbo Prolog jako różne, np.

```
wiek, wzrost = integer.
```

Użycie obiektu typu "wzrost" tam, gdzie oczekujemy obiektu typu "wiek", jest sygnalizowane przez Turbo Prolog jako błąd syntaktyczny.

● Sekcja definicji relacji (predicates section)

Wszystkie relacje używane w programie muszą być uprzednio zadeklarowane, tj. musi zostać podana liczba argumentów i ich typy, np.:

```
mieszka(string, string, integer).
```

Gdy w sekcji definicji typów użyliśmy definicji:

```
nazwisko, adres = string
telefon = integer
```

definicja relacji "mieszka" może wyglądać tak:

```
mieszka(nazwisko, adres, telefon).
```

● Opcjonalna sekcja definicji relacji między bazami danych (database section)

Użytkownik podaje tu nazwy baz danych (może być ich kilka) i fakty, które mogą być w nich przechowywane. Definicje wyglądają tak, jak w poprzedniej sekcji; różnica polega na tym, że relacje umieszczane w bazie danych możemy usuwać i dodawać (w standardowym Prologu, jak już zaznaczyłem, wszystkie relacje w programie mogą być dynamicznie zmieniane, w Turbo Prologu jest to ograniczone do sekcji bazy danych i dotyczy jedynie faktów, a nie reguł, których w tej sekcji nie można umieszczać).

● Sekcja definicji reguł (clauses section)

Ta część stanowi właściwy program w Turbo Prologu. Definiujemy tu reguły wnioskowania. Przypomina to definicje procedur w standardowych językach programowania.

● Opcjonalna sekcja dyrektyw (goal section)

Sekcja ta składa się z ciągu klauzul, których wykonania oczekujemy (odpowiednik programu głównego w językach imperatywnych). Wymagana jest jedynie przy kompilacji na programy typu EXE; przy kompilacji w pamięci bez podania dyrektywy użytkownik jest o nią pytany i może wtedy podać dowolny ciąg klauzul.

Bardzo wygodną cechą Turbo Prologu, docenianą zwłaszcza przy pisaniu dużych programów, jest możliwość pisania i testowania niezależnych modułów, a następnie łączenia ich w jeden program. Dodatkową pomocą w pisaniu dużych i efektywnych programów jest możliwość łączenia programów w Turbo Prologu z innymi językami firmy Borland, mianowicie Turbo C i Assemblerem (producent podaje nawet, że powinno być możliwe łączenie ze wszystkimi językami, które dają w wyniku kompilacji moduły typu

test COMPUTER

OBJ, choć gwarantuje poprawną pracę jedynie w przypadku dwóch wymienionych wyżej języków). Jedynym ograniczeniem jest wymóg, by program główny był napisany w Turbo Prologu, tak że np. dołączane procedury napisane w języku C, oprócz bogatej listy swoich własnych procedur bibliotecznych, mogą korzystać ze standardowych predykatów Turbo Prologu! Takie rozwiązanie jest pożyteczne nie tyle ze względu na szybkość działania programu wynikowego, bo tu Turbo Prolog w pełni zasługuje na pierwszy człon swojej nazwy, lecz na niedłączne od Prologu jego zapotrzebowanie na pamięć.

Nowa wersja Turbo Prologu, którą tu omawiam, jest dość istotnym rozszerzeniem poprzedniej (1.0). Przede wszystkim rzuca się w oczy (dosłownie) grafika, zgodna ze standardem przyjętym w Turbo Pascalu i Turbo C, tak że tworzenie aplikacji intensywnie wykorzystujących grafikę nie powoduje konieczności uczenia się nowych procedur.

Innym rozszerzeniem jest możliwość operowania na bazach danych przechowywanych na dysku, bez konieczności ich uprzedniego wczytywania do pamięci, co było konieczne w wersji 1.0. Została naturalnie zachowana możliwość operowania na bazie danych przechowywanej w pamięci lub, co znów jest nowością, w pamięci rozszerzonej (EMS). Daje to użytkownikowi dużą swobodę w dopasowaniu się do konkretnych potrzeb i wybraniu najrozsądniejszego kompromisu między wielkością bazy a czasem dostępu.

Znacznie wzrosła możliwość programowej obsługi błędów, ulepszona jest także komunikacja z użytkownikiem (udoskonalony edytor, możliwość pracy w trybie tekstowym o podwyższonej rozdzielczości dla kart EGA i VGA itp.). Pozostał jednak mało wygodny mechanizm sygnalizacji umiejscowienia ewentualnego błędu wykonania: wypisany zostaje adres, który dopiero po podaniu go w środowisku Turbo Prologu pozwala znaleźć linię w tekście źródłowym, w której błąd wystąpił.

Podsumowanie

Na koniec zadajmy sobie pytanie: do rozwiązywania jakich problemów i zadań Turbo Prolog nadaje się najbardziej? Głównym obszarem jego zastosowań jest przetwarzanie informacji symbolicznej, np. systemy ekspertowe, przetwarzanie języka naturalnego, "inteligentne" bazy danych itp., natomiast jest słabo dostosowany do obliczeń numerycznych. Częstym zastosowaniem Prologu jest tworzenie makiet systemów docelowych, a następnie, po uzgodnieniu i zaakceptowaniu systemu w zaprezentowanym kształcie, stworzenie wersji bardziej efektywnej bądź przez napisanie systemu w innym języku, bądź, jak to jest możliwe w przypadku Turbo Prologu, implementację w innym języku jedynie pewnych procedur. Zwiększona wydajność programisty przy posługiwaniu się Turbo Prologiem, szybkość działania skompilowanych programów i duża elastyczność, związana zwłaszcza z możliwościami łączenia go z innymi językami, czyni z niego bardzo sprawne narzędzie, a efekty wynikające ze stosowania go w pełni wynagradzają czas poświęcony na jego opanowanie.

Zalety Turbo Prologu wersja 2.0

- kompilacja na programy typu EXE
- możliwość pisania programów w postaci oddzielnych modułów
- duża szybkość działania zarówno kompilatora, jak i skompilowanych programów
- możliwość łączenia z innymi językami
- rozbudowana grafika
- dobry opis i duża liczba wyczerpujących przykładów

Wady Turbo Prologu wersja 2.0

- brak wbudowanych mechanizmów do metaprogramowania
- ograniczony system podpowiedzi (help)
- brak samodzielnego (stand-alone) kompilatora
- niepełna informacja o błędach wykonania

Program: Turbo Prolog wersja 2.0

Opis: kompilator Prologu rozszerzonego o typy danych

Producent: Borland International

Rok produkcji: 1988

Sprzęt: IBM PC XT/AT/386, min. 384 kB RAM (polecane 640 kB), 2 napędy dyskietek lub twardy dysk

Zawartość pakietu: 4 dyskietki (nie zabezpieczane przed kopiowaniem), 2 tomy dokumentacji (razem ok. 950 stron).

Przykładowy program w Turbo Prologu obliczający trasę między dwoma dowolnymi miastami na podstawie dróg między miastami sąsiednimi (rozszerzenie przykładu nr 2 z rozdziału 18 podręcznika użytkownika):

```
domains
  miasto = symbol
  dystans = integer

predicates
  droga(miasto, miasto, dystans)
  droga1(miasto, miasto, dystans)
  droga_inf(miasto, miasto, dystans)
  trasa(miasto, miasto, dystans)
  prog

clauses
  droga(tampa, houston, 200).
  droga(gordon, tampa, 300).
droga(houston, gordon, 100).
  droga(houston, kansas_city, 120).
  droga(gordon, kansas_city, 130).
  droga_inf(X,Y,Dystans):-
    droga1(X,Y,Dystans),!,
    write("Jedziesz z ",X," do ",Y,"."),nl.

  droga1(X,Y,Dystans):-droga(X,Y,Dystans),!.
  droga1(X,Y,Dystans):-droga(Y,X,Dystans).
  trasa(Miasto1, Miasto2, Dystans) :-
    droga_inf(Miasto1, Miasto2, Dystans).
  trasa(Miasto1, Miasto2, Dystans) :-
    droga_inf(Miasto1, Dyst1),
    trasa(X, Miasto2, Dyst2),
    Dystans=Dyst1+Dyst2, !.

  prog:-
    write("Podaj pierwsze miasto:"),readln(M1),nl,
    write("Podaj drugie miasto:"),readln(M2),nl,
    trasa(M1,M2,Odl),!,
    write("Odległość między ",M1," a ",M2," wynosi ",Odl,"
km."),nl.
  prog:-
    write("Przykro mi – nie mam wystarczających danych").
goal
  prog.
  Przykładowe działanie programu:
Podaj pierwsze miasto: kansas_city
Podaj drugie miasto: tampa
Jedziesz z kansas_city do houston.
Jedziesz z houston do tampa.
Odległość między kansas_city a tampa wynosi 320 km.
```

Komputer test



Z edytora tekstu **WordPerfect 5.0** korzystałem podczas stypendium na Wydziale Marketingu Uniwersytetu Strathclyde w Glasgow. Zetknąłem się z nim zupełnie przypadkowo.

W sobotnie popołudnie, wkrótce po przyjeździe z Polski udałem się do uniwersyteckiego centrum komputerowego, by przygotować tekst na zbliżające się seminarium. Biblioteka programów była już zamknięta. Zostałem więc ograniczony do programów zainstalowanych na twardych dyskach dostępnych komputerów. Wśród nich był tylko jeden edytor tekstu – właśnie WordPerfect. Wkrótce przekonałem się, czemu jest on najbardziej popularnym tego rodzaju programem w USA, Kanadzie i kilku krajach Europy Zachodniej.

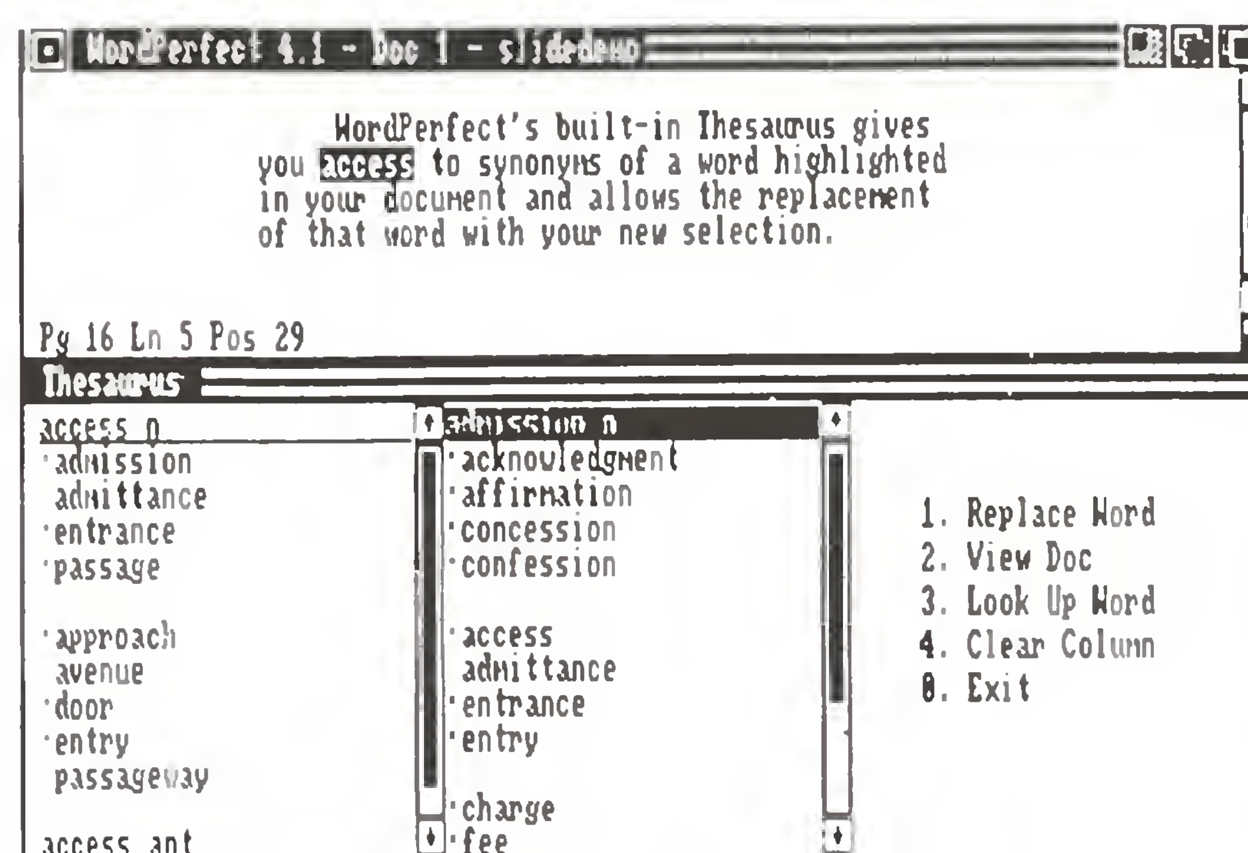
Zafascynowała mnie jego elastyczność i uniwersalizm, prostota obsługi wynikająca z precyzyjnego, logicznego sposobu skomponowania oraz profesjonalne możliwości zastosowania.

Elastyczność i uniwersalizm

Chociaż WordPerfect powstał na bardzo chłonnym rynku amerykańskim, opracowany został nie tylko z myślą o użytkownikach z krajów anglosaskich. Ze zbioru charactr.doc można wczytać dodatkowe litery języków zapisywanych w alfabecie łacińskim oraz alfabet grecki, hebrajski, japoński i cyrylicę, ponadto różne znaki typograficzne, symbole obrazkowe i 440 znaków, umożliwiających zapis operacji matematyczno-logicznych.

Przystosowanie standardowej wersji WordPerfectu do potrzeb użytkownika jest czynnością prostą dzięki ekranom edycji klawiatury i klawiszy. Procedury makro pozwalają na dowolne zaprogramowanie każdego klawisza bądź ich kombinacji. W ten sposób tworzymy, w zależności od potrzeb i przyzwyczajeń, "własne" klawiatury (np. polską, niemiecką, rosyjską, matematyczną, bankową etc.). W moim przypadku zaprogramowanie polskich liter zajęło mi około 10 minut. Teraz po zainstalowaniu "polskiej" klawiatury **(SHIFT+F1, 6,** najechanie kursorem na obrazek "mojej" klawiatury), kombinacja **CTRL+x** pozwala na uzyskanie wielkich, natomiast **ALT+x** małych "polskich" liter (x oznacza odpowiednią literę łacińską np. a, e, c itd.).

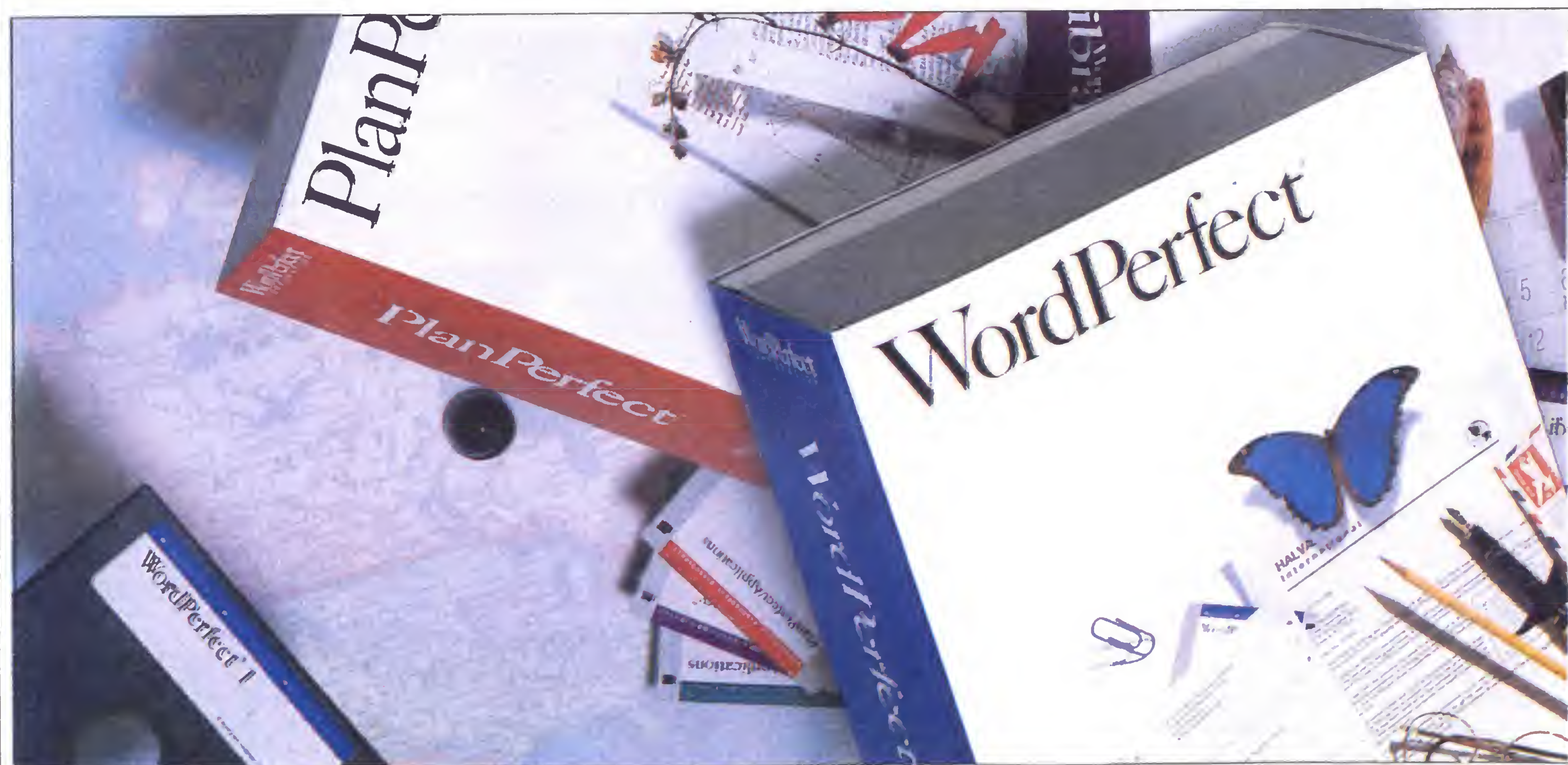
Autorzy edytora zadbali o jego zastosowanie w skali międzynarodowej. Jeżeli zamierzamy pisać tekst w języku francuskim, hiszpańskim, włoskim, portugalskim, niemieckim, islandzkim, holenderskim, norweskim, fińskim lub szwedzkim wystarczy przy określaniu formatu dokumentu **(SHIFT+F8)** podać kod języka, a będą stosowane reguły przenoszenia wyrazów oraz zapisywanie dat we właściwy sposób. Wprawdzie w wyposażeniu podstawowym są słowniki, ortograficzny i wyrazów bliskoznacznych "jedynie" języka angielskiego, jednak można kupić dyskietki ze słownikami w wymienionych językach.



Słownik synonimów. Wersja dla Amigi.

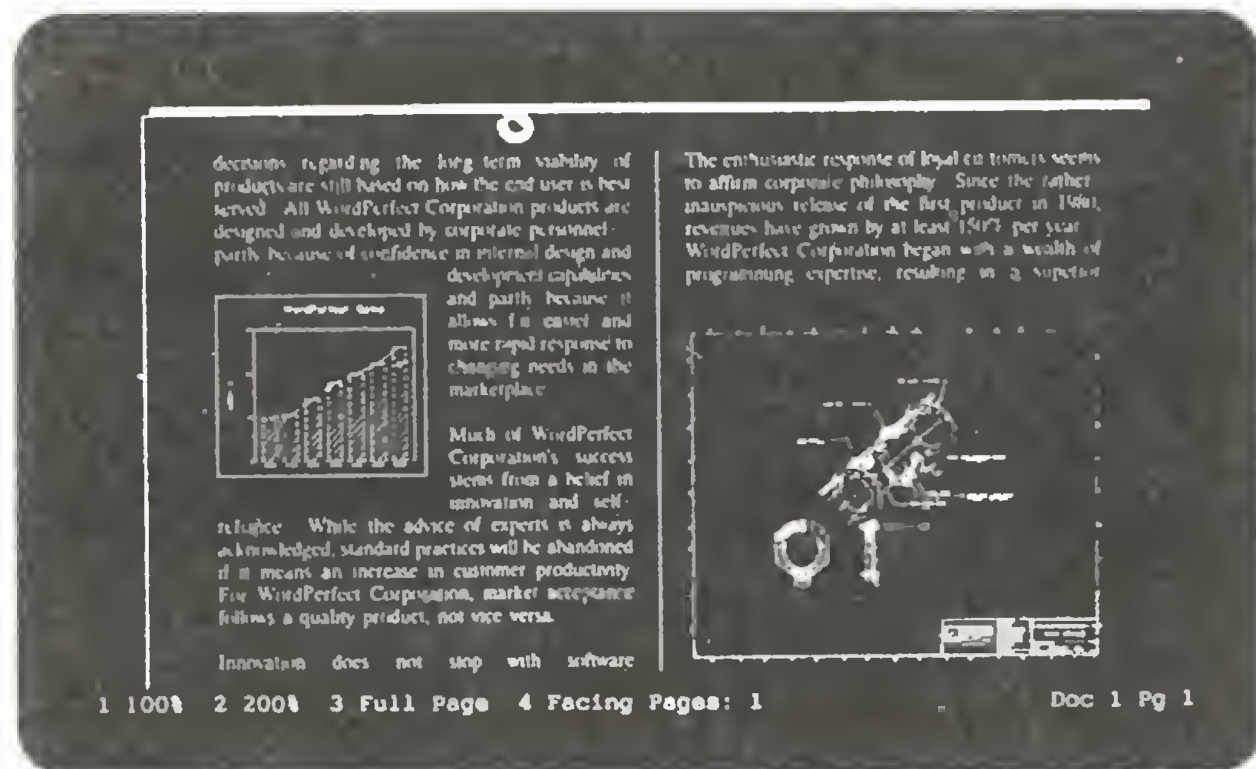
Ostre wymogi jakie stawiali przed sobą autorzy tego pakietu, by nadać mu charakter uniwersalny, widać na przykładzie współpracy z drukarkami. Na czterech dyskietkach są programy przygotowujące do wydruku dokumenty na blisko 100 drukarek i błyskawicznie przeformatowujące teksty na drukarkę, którą mamy podłączoną do komputera. Zapewne i ta cecha WordPerfecta sprawia, że jest on – jak się przekonałem podczas mojego stypendium – wykorzystywany przez wielu stypendystów. Gwarantuje on, że z przygotowanymi w jednym miejscu tekstami nie będzie gdzie indziej problemów z wydrukiem.

Edytor może pracować z kolorowym monitorem. Standardowy ekran roboczy to niebieskie tło i białe litery. Jeżeli użytkownik uzna, że to zbyt duży kontrast, może to zmienić w niemal dowolny sposób. Sekwencja **SHIFT**+F1,3,2,1 umożliwia w zależności od typu monitora oraz tego czy wykorzystujemy jedynie znaki ASCII czy też dodatkowe, na wybór tła z 16 lub 8 kolorów oraz na zmiany w kombinacji barw do oznaczania wielkości i wyglądu fragmentów dokumentu.



Możliwa jest wzajemna konwersja dokumentów przygotowanych w WordPerfect i innych programach. Bezpośrednia konwersja zachodzi między WordPerfectem 5.0 a jego wersją 4.2 oraz **WordStarem 3.3** i **Multimate 3.22**. W innych przypadkach konwersja jest dwustopniowa. Wykorzystywane są do tego pakiety formatujące *Revisable-Form-Text* (DCA), arkusz kalkulacyjny DIF (*Data Interchange Format*) i *Mail Marge Format*. Umożliwiają one współpracę WordPerfecta z dokumentami powstałymi za pomocą programów: **DisplayWrite 3, Lotus i dBase**. Konwersja arkusza kalkulacyjnego w Lotusie 1-2-3 na dokument w WordPerfect wymaga najpierw (w Lotusie) konwersji na arkusz kalkulacyjny DIF, który jest następnie przez program convert przekształcany na dokument WordPerfecta.

Jeśli potrzeba, w prosty sposób łączymy tekst z grafiką. WordPerfect "wciąga" wykresy powstałe na bazie 32 programów graficznych. Z najbardziej znanych to **Lotus 1-2-3, Harvard Graphics Presentation, PC Paintbrush, SuperCalc 4** czy **HP Graphics Gallery**. Konwersji grafiki, podobnie jak tekstów, dokonuje się z poziomu DOSu (można chwilowo wejść w system operacyjny – **CTRL+F1,1**).



Wersja dla IBM.

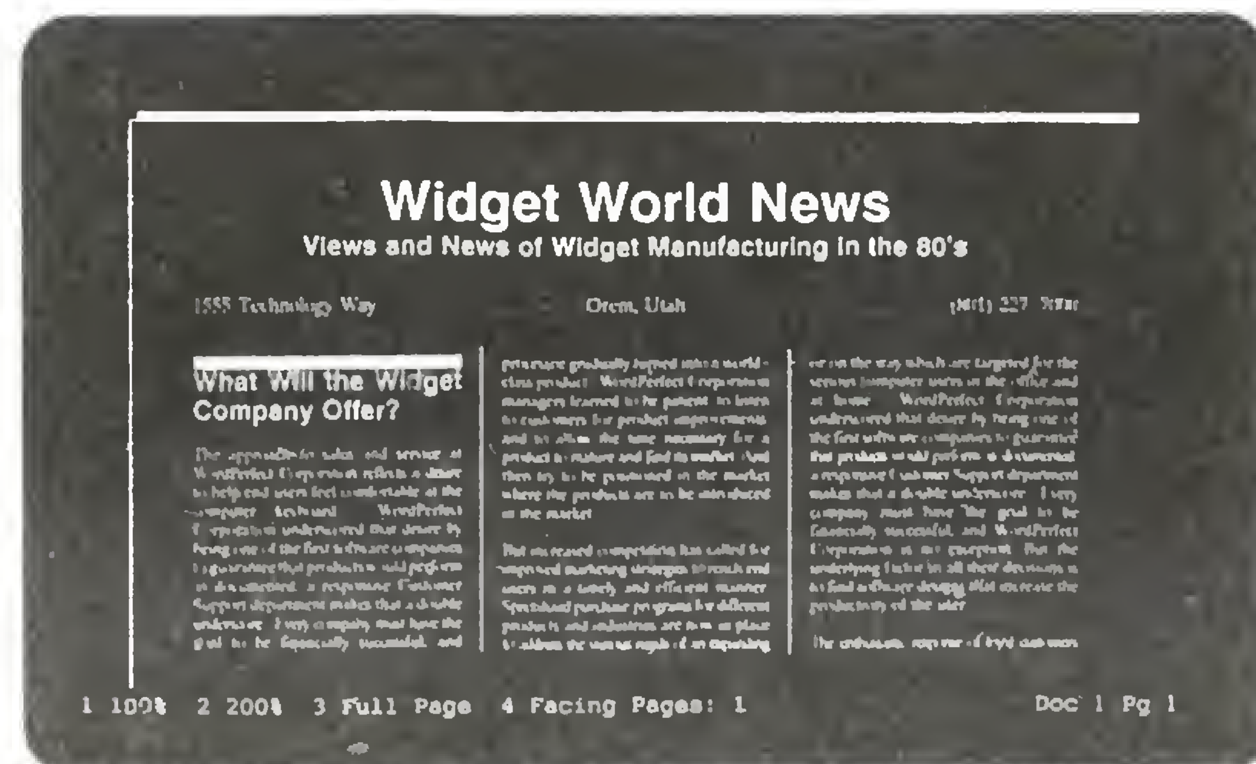
Prostota obsługi

Kluczem do szybkiego opanowania WordPerfecta jest bardzo dobry samouczek **(F3)**. Pozwala on na przypomnienie sobie sekwencji komend niezbędnych do wykonania określonej operacji, bądź na zapoznanie się z obszernym opisem wybranej funkcji wraz z opcjami.

Bardzo pomocna do przenoszenia lub kopiowania kolumn tablic jest funkcja bloku (**ALT+F4**), która działa także pionowo.

WordPerfect ma dwa ekrany robocze, które pozwalają na pracę na dwóch dokumentach jednocześnie. Opcja ekranu ujawnionych kodów (**ALT+F3**) pokazuje wszystkie znaki sterujące użyte przy pisaniu tekstu. Gdy kompilujemy dokument z wcześniej powstałych zbiorów, zdarza się, że nie przybiera on postaci przez nas oczekiwanej. Skorzystanie z podglądu w opcji ekranu ujawnionych kodów ułatwia szybkie znalezienie i wymazanie niepotrzebnych znaków sterujących.

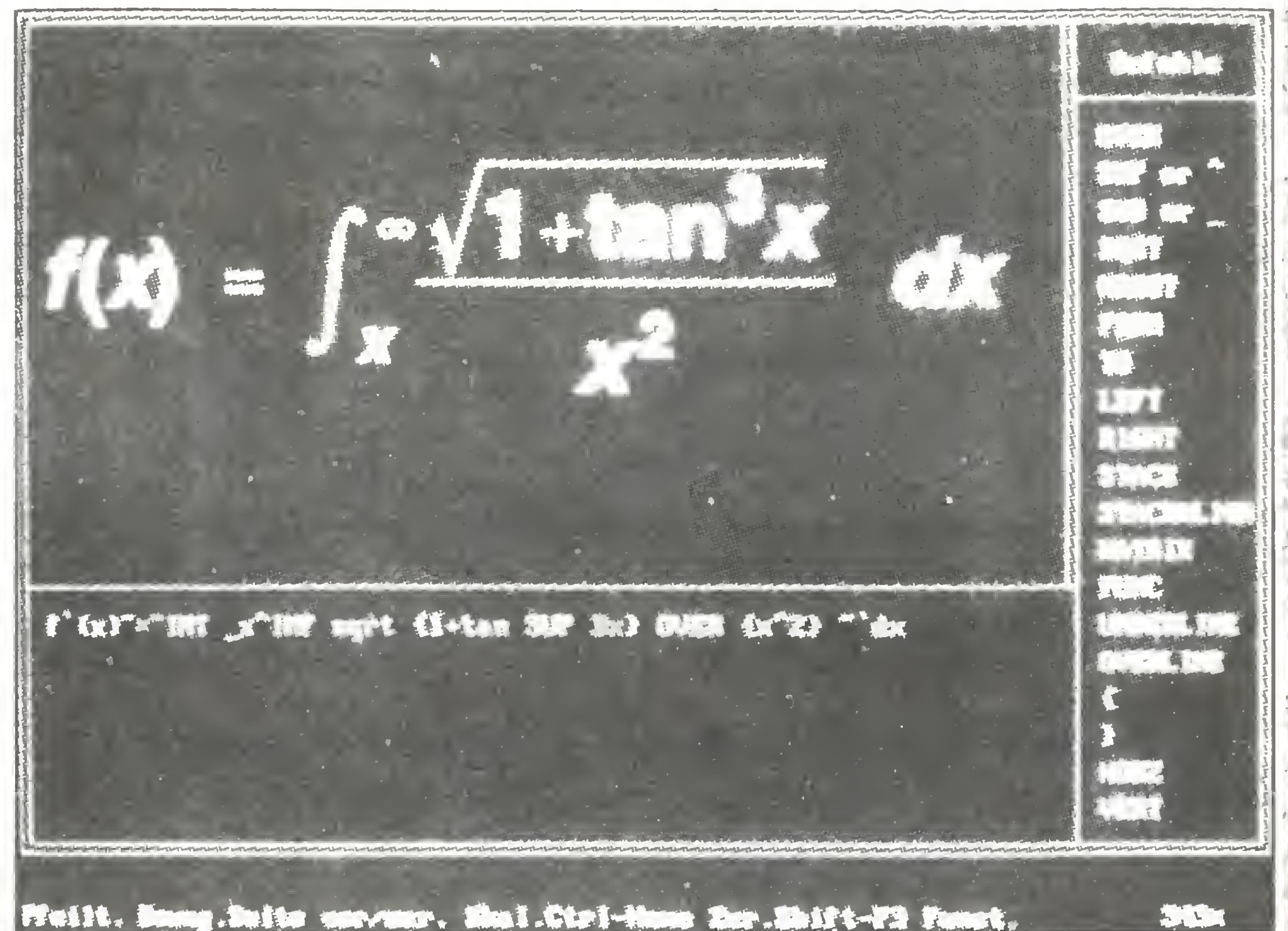
WordPerfect jest bardzo szybki, gdyż nie pracuje w trybie graficznym. Gruby druk, podkreślenia, małe bądź wielkie litery zaznaczane są różną kombinacją kolorów. Jeżeli przed wydrukowaniem chcemy zobaczyć, jak nasze dzieło będzie się prezentować, wówczas przechodzimy w tryb graficzny i mamy na monitorze wygląd całych stron tekstu lub w powiększeniu dowolnych jego fragmentów **(SHIFT+F7,6)**.



Wersja dla IBM.

Możemy też opatrywać tworzony dokument komentarzem **(CTRL+F5,5)**. Pozwala to w dowolnym momencie przerwać pracę nad dokumentem i zapisać na jego tle nasuwające się uwagi, które później mogą być przydatne. Zapisany komentarz nie jest drukowany i jest oddzielany podwójną ramką od zasadniczego tekstu, co ułatwia jego wychwycenie.

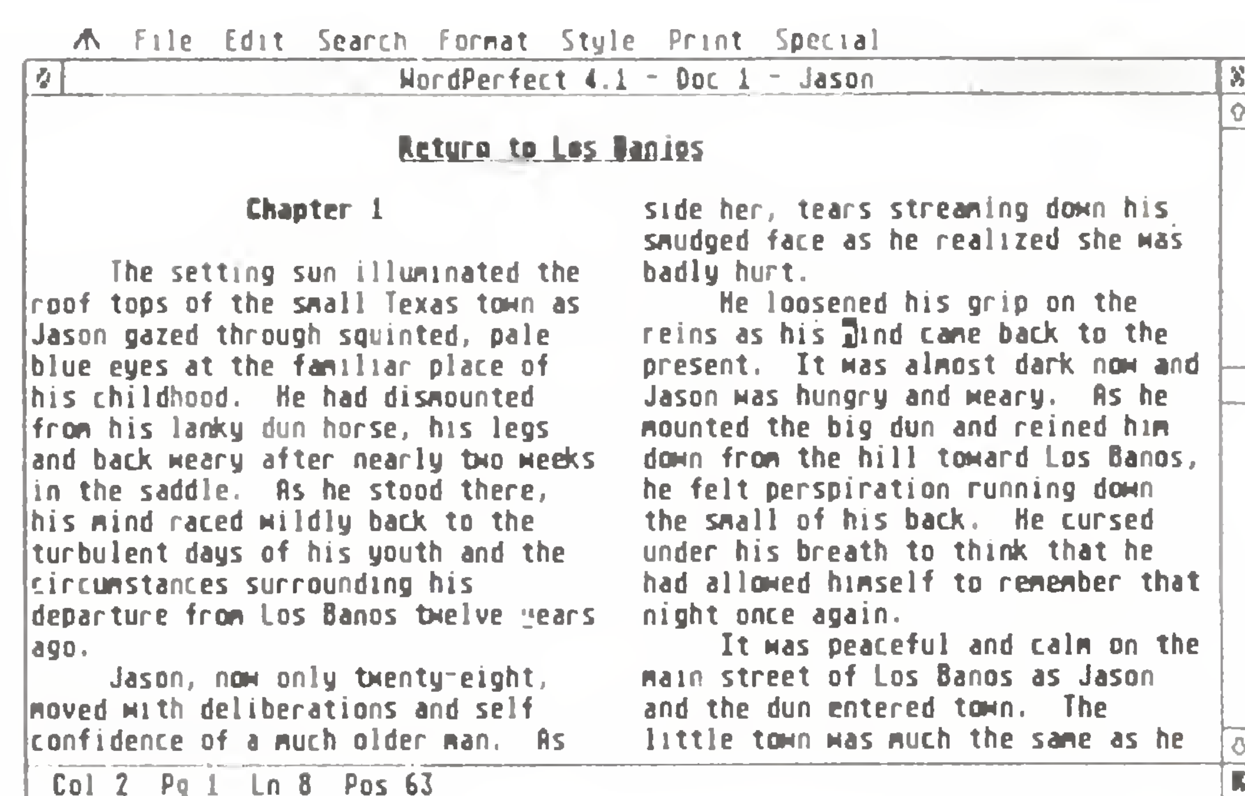
Zaletą edytora jest dokonywanie operacji matematycznych oraz, gdy chcemy, zabezpieczanie dokumentu przed innymi.



Zaawansowani użytkownicy mogą stosować rozbudowane procedury makro. Można w nich używać, oprócz innych, także rozkazy warunkowo-logiczne takie jak: **IF-ELSE, IF EXISTS, GO, PAUSE** oraz tworzyć pętle w stosunku do podstawowego programu. Przyśpiesza to pracę edytora.

Mankamentem WordPerfecta jest przypisanie klawiszowi **Esc** innego znaczenia niż w większości programów. Nie służy on do wyjścia z rozwiniętego menu, lecz powtórzeniu określonych funkcji n razy. Funkcję "tradycyjnego" **Esc** wykonuje klawisz funkcyjny **F1**. Oczywiście używając procedury makro można to zmienić.

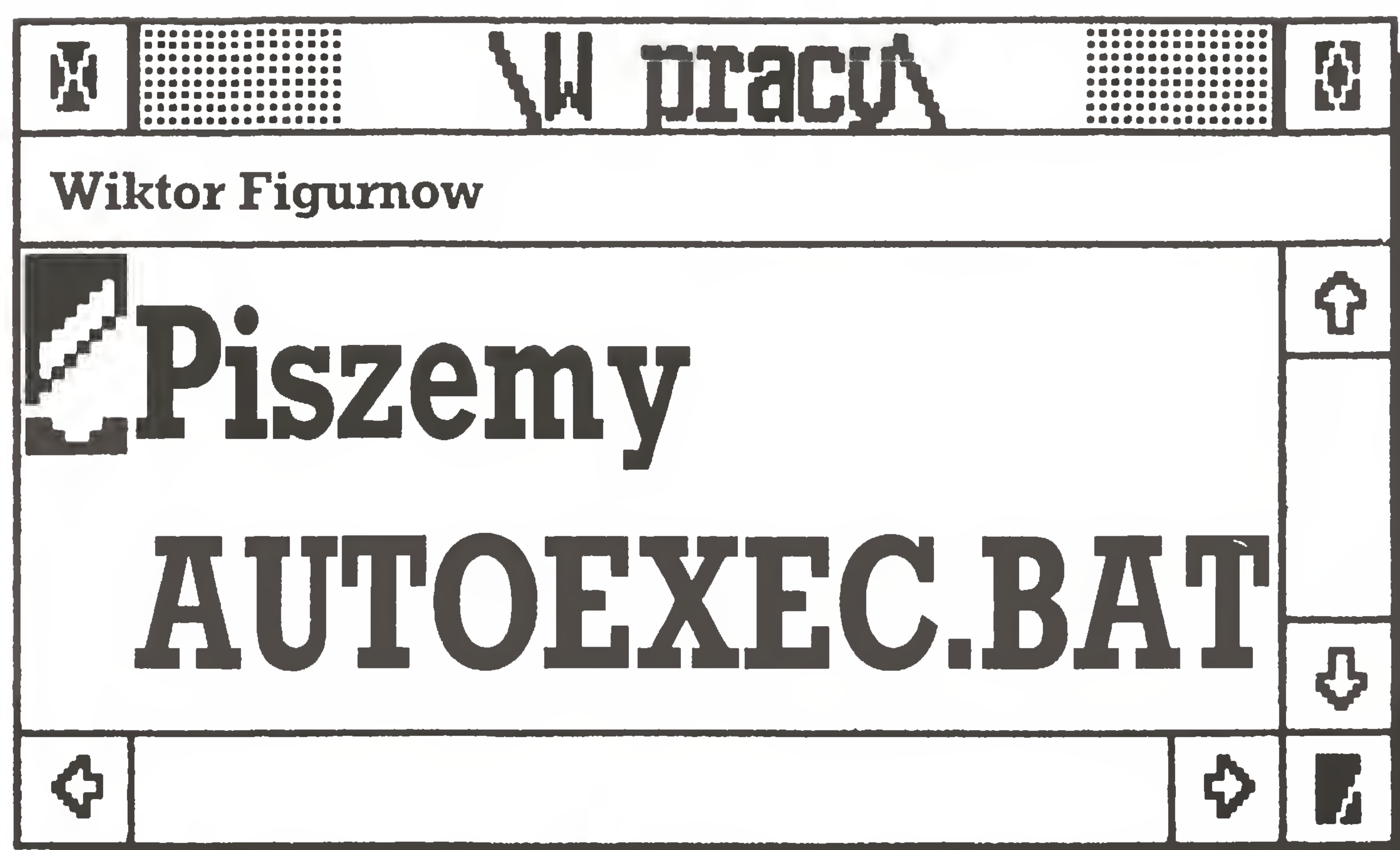
Jeśli chcemy, możemy pisać tekst w szpaltach na wzór gazet **(ALT F7,4)**. Dodatkowo możemy je dowolnie łamać, by wmontować w tekst wykresy lub rysunki.



Szpaltowanie Wersja dla ATARI ST.

Pakiet WordPerfect 5.0 zawiera 12 dyskietek instalacyjnych i podręcznik użytkownika. Polecam go głównie tym, którzy piszą skomplikowane teksty i chcą mieć efektywnie pracujący edytor z elementami programu typu DTP.





Kontynuujemy rozpoczęte w poprzednim numerze "Komputera" rozważania o tym, jak stworzyć w IBM PC najdogodniejsze środowisko pracy. Tym razem przygotujemy wzorcowy AUTOEXEC.BAT.

Plik o tej nazwie powinien się znajdować w katalogu głównym dysku (dyskietki), z którego ładowany jest system operacyjny. Jest on wykonywany automatycznie po załadowaniu systemu. Do AUTOEXECA wpisywane są polecenia, które mają być wykonywane przez komputer zaraz po załadowaniu systemu operacyjnego. Za ich pomocą można "skonfigurować" system stosownie do potrzeb użytkownika. Ponadto w przypadku znalezienia na dysku pliku o tej nazwie system operacyjny nie pyta podczas ładowania o bieżącą datę i godzinę. Samo słowo AUTOEXEC oznacza "samowykonujący się".

Przygotowanie spisu katalogów, w których poszukiwane będą wykonywane programy

DOS-owe polecenie PATH, które powinno znaleźć się w pliku AUTOEXEC.BAT, jest wykazem katalogów, w których poszukiwane będą wykonywane programy. Wiadomo, że w takim przypadku polecenie uruchomienia danego programu można wprowadzać znajdując się w dowolnym katalogu, a nie koniecznie w tym, w którym jest sam program. Polecenie ma następującą składnię:
PATH = {wykaz katalogów oddzielonych średnikami}
Przykład: PATH = C:\;C:\EXE;C:\EXE\NORTON;..;..\;..

W tym przykładzie uruchamiany program (któryś z programów użytkowych z pakietu **Norton Utilities**) będzie poszukiwany kolejno: w katalogu głównym, podkatalogu EXE, podkatalogu katalogu EXE o nazwie NORTON. Celowe jest również wskazanie w AUTOEXECU ścieżek dojścia do katalogów .. i podkatalogów pierwszego i drugiego stopnia katalogów, wymienionych w poleceniu PATH.

Pamiętajmy, iż długość wiersza rozkazowego w systemie operacyjnym MS-DOS wersji poniżej 3.3 jest ograniczona, stąd zbyt obszerna lista katalogów wymienionych w poleceniu PATH może się po prostu nie zmieścić.

Ustalenie wzorca zachęty systemowej MS-DOS

Za pomocą polecenia Prompt zmieniamy wzór zachęty systemowej, świadczącej o gotowości systemu operacyjnego do wykonywania poleceń.
Format polecenia: Prompt {tekst wzorca}

W tekście wzorca zachęty systemowej stosujemy specjalne znaki, mające następujące znaczenia:

$p – bieżący katalog i dysk
$n – domyślny katalog
$d – bieżąca data
$t – bieżący czas
$h – usunięcie poprzedniego znaku (DOS 3.3)
$s – spacja
$e – znak o kodzie 27 – ESC (początek sekwencji ANSI)
$g – znak >
$l – znak <
$b – nowy wiersz
$v – numer wersji systemu operacyjnego

Pozostałe wymienione we wzorcu znaki pojawiają się w zachęcie bez żadnych modyfikacji.

Najczęściej używane wzorce zachęty systemowej mają następującą postać:

prompt pg – zachęta systemowa zawiera informację o bieżącym dysku i katalogu oraz znak ">" (na przykład C:\EXE\NORTON>);

prompt $t pg – j.w. oraz dodatkowo informacja o bieżącym czasie (np. 13:05:43.56 C:\EXE\NORTON>).

Wielu użytkowników uzna taki wzorzec zachęty za zbyt długi. Aby go skrócić należy przy pomocy polecenia $h usunąć z tekstu zachęty systemowej informacje o sekundach i ułamkach sekund. Jeśli więc tekstowi wzorca nadamy postać **prompt thhhhh$h pg** systemowa zachęta przybierze postać 13.05 C:∧EXE∧NORTON>.

Jeśli podczas uruchamiania systemu operacyjnego potrzebny nam będzie driver ANSI.SYS (dla przypomnienia – informacja o tym powinna znajdować się w pliku CONFIG.SYS w postaci device = {imię katalogu\ansi.sys}), to we wzorcu zachęty można wykorzystać dodatkowo różne kolory poszczególnych znaków i tła, na jakim będą one umieszczone. W tym celu w poleceniu Prompt wskazujemy atrybuty drivera ANSI.SYS:

$e[<atrybut>;...<atrybut>m

Wartość poszczególnych atrybutów ma postać liczb dziesiętnych. Jeśli w poleceniu wymieniamy więcej niż jeden atrybut, należy je rozdzielić średnikami.

Znaczenia atrybutów:
0 – znaki standardowe (białe na czarnym tle);
1 – znaki bardziej jaskrawe;
5 – znaki pulsujące;
7 – znaki w inwersie (czarne znaki na białym tle);
8 – znaki niewidczne (kolor znaku pokrywa się z kolorem tła);
30-37 – kolor znaku czarny, czerwony, zielony, brązowy, granatowy, niebieski lub biały.
40-47 – kolory tła czarny, czerwony, zielony, brązowy, granatowy, niebieski lub biały;

Kolory można na siebie nakładać. Tak więc, by przykładowo znaki w zachęcie systemowej były żółte, polecenie powinno mieć postać **$e[33;1m.**

Zwolenników napisów wielokolorowych zachęcam do eksperymentów estetycznych. Przykładowo polecenie **prompt $e[7m$p$e[0m$g** spowoduje wyświetlenie informacji o bieżącym katalogu w inwersie (na monitorze monochromatycznym), zaś pozostałego tekstu zachęt – normalnie, biało na czarnym. Polecenie **prompt $e[36m$thhhhhh $e[35m$p$e[32m$g$e[0m** spowoduje wyświetlenie informacji o bieżącym czasie w kolorze niebieskim, bieżącym katalogu – w kolorze wrzosowym, zaś znaku ">" – w kolorze zielonym.

Za pomocą poleceń drivera ANSI.SYS można umieszczać znaki zachęty w różnych miejscach ekranu. Przykładowo, informację o bieżącym czasie można umieścić w prawym górnym rogu ekranu. W tym przypadku treść wzorca zachęty systemowej powinna zawierać następujące polecenia: **$e[s** – zapamiętanie bieżącej pozycji kursora; **$e[u** – umieszczenie kursora w miejscu określonym wcześniej poleceniem **$s[s; oraz $e[**<wiersz>;<kolumna>H – umieszczenie kursora w określonym miejscu ekranu.

Tak więc polecenie **$e[s$e[1;52H$e[44;33;1m $t $d $e[0;35m$e[upe[32mge[0m** spowoduje wyświetlenie informacji o bieżącej dacie i czasie w prawym górnym rogu ekranu, żółtymi cyframi na granatowym tle, zaś informacji o bieżącym katalogu (kolorem wrzosowym) i znaku ">" (zielonego koloru) w bieżącym wierszu.

Określenie zmiennych środowiska

System operacyjny może przeznaczyć wyodrębnioną część pamięci operacyjnej, zwaną "środowiskiem" (Environment) dla przechowywania w niej wartości niektórych zmiennych, wykorzystywanych przez system operacyjny i inne programy. Zawartość wspomnianego środowiska ma postać "zmienna = wartość", gdzie zmienna – to dowolny wiersz nie zawierający znaku "=". W zapisie wartości małe i wielkie litery nie są rozróżniane. Wartość – to dowolny ciąg znaków.

System operacyjny MS-DOS wykorzystuje trzy umieszczone w CONFIG.SYS zmienne środowiska: Path, Prompt i Comspec, które określane są odpowiednimi poleceniami Path, Prompt i Command z parametrem /P. Użytkownik może określić wartość zmiennych środowiska poleceniem Set. Format polecenia: Set zmienna = wartość (np. Set CHIFILES = C:\CHI).

Jeśli wymienionej w poleceniu zmiennej nadana została już wcześniej jakaś wartość – zostaje ona zastąpiona nową.

Programy użytkowe umożliwiają analizę zawartości pamięci, w której przechowywane są zmienne środowiska, oraz określanie i modyfikowanie ich wartości. Zazwyczaj zmienne środowiska wykorzystywane są po to, by określić gdzie poszczególne programy użytkowe powinny szukać swych pomocniczych plików.

Przykład zastosowania polecenia, tym razem ze sfery antywirusowej. Jak wiadomo, najczęściej zarażany bywa umieszczony w katalogu głównym COMMAND.COM. Aby zredukować groźbę zara-

żenia tego pliku, należy go umieścić w dowolnym katalogu, poinformować o tym system, dodając do pliku CONFIG.SYS wiersz SHELL=C:\{nazwa katalogu}^COMMAND.COM /P, po czym wykorzystać wspomniane zmienne umieszczonym w AUTOEXEC.BAT poleceniem Set Comspec=C:\{nazwa katalogu}\COMMAND.COM.

Automatyczne uruchomienie programów

Do pliku AUTOEXEC.BAT można włączyć polecenia automatycznego uruchomienia programów, z których korzystamy podczas pracy z innym oprogramowaniem (np. drivery znaków narodowych – ekranowe i klawiaturowe), czy też nakładek na system operacyjny. Skoncentrujmy się na najważniejszych z nich.

● **Drivery klawiatury.** Aby umożliwić wprowadzanie z klawiatury znaków narodowych, należy odwołać się do pomocy specjalnych programów, konfigurujących klawiaturę stosownie do potrzeb. Programy takie mogą być uruchamiane automatycznie wraz z uruchomieniem komputera, o ile oczywiście informacja o tym zawarta została w pliku AUTOEXEC.BAT (lub w przypadku niektórych programów w pliku CONFIG.SYS), oraz pozostawać na stałe w pamięci operacyjnej maszyny. Po uruchomieniu takiego programu litery narodowe można wprowadzać przy pomocy odpowiedniej kombinacji klawiszy (zazwyczaj z ALT lub CTRL) lub po przełączeniu klawiatury na inny alfabet (np. cyrylicę).

● **Drivery ekranu.** Stosowane rzadziej, z reguły w przypadku maszyn z kartą EGA, VGA etc. umieszczają w pamięci ekranowej dodatkowy zestaw znaków narodowych. Współdziałanie obydwu driverów (niekiedy oba zawarte są w jednym programie) umożliwia wyświetlanie wprowadzanych z klawiatury znaków narodowych.

● **Drivery drukarek.** Stosowane jeszcze rzadziej, umożliwiają download znaków narodowych do bufora pamięci drukarki i drukowanie tekstów z takimi znakami w trybie tekstowym, a więc znacznie szybciej niż w trybie graficznym.

● **Programy umożlwiające natychmiastowe wydrukowanie kopii ekranu.** Jeden z tych programów – to systemowy Graphics z zestawu programów MS-DOS. Umożliwia wydruk kopii ekranu na drukarce zgodnej programowo z IBM Color Printer lub IBM Graphics Printer (większość popularnych w Polsce drukarek umożliwia pracę w jednym ze wspomnianych trybów, należy jedynie zmienić ustawienie mikroprzełączników), o ile ekran pracuje w trybie zgodnym z jednym z trybów graficznych karty CGA. Istnieją programy, umożliwiające wydruk kopii ekranów pracujących w innych trybach i na innych drukarkach.

● **Programy do ustawienia systemowego zegara w komputerze.** W pierwszych modelach "pecetów" zegar systemowy ustawiać trzeba było ręcznie po każdym uruchomieniu komputera. Niedogodność ta została z czasem usunięta przez producentów, którzy umieścili w komputerach specjalny układ zawierający zegar z zasilającą go baterią lub akumulatorem, która podtrzymywała jego pracę, kiedy komputer był wyłączony. Umieszczony w AUTOEXEC.BAT program (np. TIMER czy REALTIME) umożliwia odczytanie czasu po włączeniu komputera i zapamiętanie go przez system operacyjny, który od tej chwili przejmuje kontrolę nad pracą zegara.

W komputerach IBM PC/AT i PS/2 odczyt informacji z wbudowanego zegara odbywa się automatycznie i umieszczanie w AUTOEXEC.BAT polecenia uruchomienia wspomnianych programów jest niecelowe.

● **Program DOSEDIT.** Przechowuje wprowadzane z klawiatury polecenia i umożliwia ich wypisanie w bieżącej linni rozkazowej po naciśnięciu klawiszy [strzałka w górę] i [strzałka w dół]. Polecenia napisane nieprawidłowo, po których pojawia się komunikat systemowy "Bad command or file name", nie są przez program zapamiętywane. Te same funkcje co DOSEDIT pełnią inne programy, które można włączyć do AUTOEXEC.BAT np. CED.

● **Programy monitorujące** próby zainfekowania systemu lub dysków wirusami oraz

● **przeprowadzające antywirusową diagnostykę** oprogramowania. Najpopularniejsze z nich, zazwyczaj umieszczane w AUTOEXEC.BAT – to program monitorujący FLUSHOT+ i diagnozująco-leczący MKS_ VIR. Polecenie urochomienia tych programów powinno znajdować się w pliku AUTOEXEC.BAT jako ostatnie.

● Często do AUTOEXEC.BAT włączane są polecenia uruchomienia programów, wyświetlających na ekranie włączonego komputera różnego rodzaju przypomnienia np. o zaplanowanych na dany dzień spotkaniach czy pracach.

● **Nakładki na system operacyjny**, w tym najpopularniejszy Norton Commander oraz XTREE. Jeśli przyzwyczailiście się do pracy z tymi nakładkami, umieszczenie w AUTOEXEC.BAT polecenia uruchomienia takiego programu natychmiast po włączeniu komputera jest jak najbardziej celowe.

Wszystkie wymienione programy uruchamiane będą po umieszczeniu w poszczególnych wierszach AUTOEXEC.BAT nazw danych programów (np. NC, TIMER, KLAWPOL etc.).

Przeglądanie poczty

Jeśli z jednego komputera korzysta wielu użytkowników, celowe jest umożliwienie im przekazywania sobie nawzajem różnego rodzaju informacji. Przytoczony przykładowy AUTOEXEC.BAT powoduje wyświetlenie na ekranie komputera po jego włączeniu informacji zawartej w pliku tekstowym MAIL, umieszczonym w katalogu C:\DOC, o ile oczywiście plik taki został tam wcześniej umieszczony.

```
if not exist C:\DOC\MAIL goto nomail
cls
type C:\DOC\MAIL
pause :nomail
```

Macro

Wielu użytkowników wykorzystuje możliwości drivera ANSI.SYS i przypisuje poszczególnym klawiszom klawiatury różną treść, która pojawia się na ekranie po jego naciśnięciu.

Jeśli dodatkowo zawarte w driverze ANSI.SYS polecenie przypisujące określoną treść danemu klawiszowi kończy się znakiem "koniec wiersza" (znak o wartości 13), to będzie to równoznaczne z wykonaniem zadanej treści jako polecenia systemu operacyjnego.

Aby po naciśnięciu jednego klawisza w wierszu rozkazowym pojawiła się przypisana mu treść, należy do driveru ANSI.SYS wpisać polecenie <kod_klawisza>; "polecenie>". Znak> " oznacza kod 27 (Escape). Jeśli po uruchomieniu systemu naciśniemy opisany w ANSI.SYS klawisz, w wierszu rozkazowym pojawi się zapisana w driverze treść, przypisana temu klawiszowi.

Jeśli polecenie ma być automatycznie wykonane, bez konieczności dodatkowego naciskania klawisza ENTER, należy wpisane w ANSI.SYS polecenie zakończyć kodem 13, czyli nadać mu postać ANSI.SYS: <kod_klawisza>;"<polecenie>";13p.

Z reguły zadana treść przypisywana jest następującym klawiszom i kombinacjom klawiszy:

[F1] – [F10] (kody 0;59 – 0;68), [Shift-F1] – [Shift-F10] (kody 0;84 – 0;93), [Ctrl-F1] -[Ctrl-F10] (kody 0;94 – 0;103) [Alt-F1] – [Alt-F10] (kody 0;104 – 0;113).

Przyładowy AUTOEXEC.BAT

Aby treść przytoczonego wzorcowego AUTOEXEC.BAT była czytelna, opatrzyłem go komentarzami, umieszczonymi po poleceniu rem. Jeśli więc Czytelnicy zechcą z niego skorzystać, mogą wszystkie wiersze, zaczynające się od słowa rem, pominąć przy przepisywaniu.

```
@echc off
rem Określenie cieżek dojścia do używanych programów
rem
path C:\;C:\EXE\DOS;C:\EXE;C:\EXE\NORTON;..;..\..
rem
rem Określenie wzorca zachęty systemowej MS-DOS
rem
prompt $e[36m$t$h$h$h$h$h$h $e[35m$p$e[32m$g$e[0m
rem
rem Określenie zmiennych środowiska
rem
set CHIFILES=C:\CHI
set Comspec=C:\DOS\COMMAND.COM
set LIB=C:\FORTRAN;C:\MSC\LIB
set INCLUDE=C:\MSC\INCLUDE
set TMP=:E\
rem Uruchomienie drivera klawiatury
C:\EXE\FILOAD\RK
rem Uruchomienie programu DOSEDIT
DOSEDIT
rem Diagnostyka obesności wirusów
rem C:\EXE\MKS_VIR C:
rem
rem Uruchomienie monitora antywirusowego
rem
C:\EXE\FLUSHOT
rem
rem Uruchomienie programu Norton Commander
rem
NC
```

Oczywiście wszystkie opisane tu katalogi są przypadkowe z punktu widzenia Czytelnika i zostały zaczerpnięte z dysku twardego, z którym pracuję. Dostosowanie treści AUTOEXEC.BAT do realnych danych pozostawiam już Czytelnikom.

Tłumaczenie Halina Madejczyk





Przedstawiam krótki opis wirusów komputerowych z mojej kolekcji wraz z informacją o dostępnych w redakcji miesięcznika "Komputer" programach zwalczających je. Programy te dostępne są również w redakcyjnej sieci FIDO.

Wirusy 507/534

Dwie wersje tego samego wirusa, znanego również po nazwą W13. Doklejają się na końcu programów typu *.COM, zwiększając ich wielkość o 507 lub 534 bajtów. Przy każdym uruchomieniu swego nosiciela starają się zarazić jeden program. Po zarażeniu pozostawiają swój "odcisk palca" w postaci zmiany miesiąca daty ostatniej modyfikacji zbioru na 13 (wirus 534) lub 12 (507). Korzystają potem z niego, aby powtórnie nie zarazić tego samego programu. Programy do zarażenia poszukiwane są wyłącznie w aktualnym katalogu.

Jak widać z powyższej charakterystyki, wirusy te należą do "łagodnych", zarówno ze względu na ograniczony zasięg rozmnażania, jak i brak działania destrukcyjnego. Tym nie mniej, zwiększenie rozmiarów zarażonych programów prowadzi do niepotrzebnego ograniczenia pojemności dysku, zwykle i bez tego zbyt małej w stosunku do naszych potrzeb.

Jako ciekawostkę podać można tekst "Copyright Microsoft" znajdujący się w ciele obu wirusów.

Wirus 648

To chyba pierwszy wirus który dotarł do Polski. Znany jest również pod nazwami Vienna, DOS 62 i (wyłącznie w Polsce) jako 'polski'. Dokleja się na końcu programów typu *.COM, zwiekszając ich wielkość o 648 bajtów. Przy każdym uruchomieniu swego nosiciela poszukuje niezarażonego jeszcze programu, a gdy taki znajdzie, sprawdza stan zegara systemowego, traktując go jako generator liczb przypadkowych z zakresu 1 – 8. W siedmiu na osiem statystycznych przypadków znaleziony program jest zarażany. W pozostałych – wirus niszczy 5 pierwszych bajtów programu – zastępowane są one rozkazem bezwarunkowego skoku do procedury BIOS'u dokonującej przeładowania systemu operacyjnego. Po zarażeniu pozostawia swój "odcisk palca", zmieniając liczbę sekund czasu ostatniej modyfikacji zbioru na 62.

Program do zarażenia poszukiwany jest w aktualnym katalogu, następnie w katalogach zdefiniowanych dyrektywą DOS'u PATH.

Wirus CHOINKA

Ten wirus – to wynik rozpowszechniania listingów wirusów w prasie i książkach. W ten sposób pozbawionym inwencji amatorom umożliwia się dołączenie do kręgu tworców wirusów. Niewątpliwie tak było i w tym przypadku. Twórca tego stworka "wklepał" listing wirusa 648 nieznacznie go modyfikując.

Te zmiany to:
– oznaczanie zarażonych zbiorów liczbą sekund 60 (a nie 62);
– usunięcie opcji niszczenia co ósmego (statystycznie) zbioru;
– dołożenie funkcji sprawdzającej datę bieżącą i w wypadku, gdy mamy do czynienia z dniem póżniejszym niż 19 grudzień lub wcześniejszym niż 1 styczeń (nie wiem jak to sobie autor wirusa wyobrażał) wyświetlana jest choinka oraz życzenia świąteczne.

Z powodu dużych rozmiarów obrazka (cały ekran), wielkość wirusa wzrosła do 1881 bajtów.

Wirus LEHIGH

Ten dosyć prosty, lecz bardzo złośliwy wirus zaraża wyłącznie zbiór COMMAND.COM. Składa się na niego około 300 bajtów kodu, które nakładane są na zawarty w tym programie obszar przeznaczony na stos. W związku z tym zarażony program nie zmienia długości. Po każdym uruchomieniu programu COMMAND.COM (co towarzyszy wykonaniu dowolnej rezydującej instrukcji DOS'u – np. TYPE, COPY, DIR itp.) wirus sprawdza, , czy na dysku będącym obiektem instrukcji znajduje się zbiór COMMAND.COM. Jeżeli tak – zaraża go, po czym inkrementuje zawarty w swoim ciele licznik. W wypadku, gdy wartość tego licznika osiągnie 4, wirus kasuje pierwsze 32 sektory wszystkich dołączonych do komputera dysków. Praktycznie, niszczy to informacje na dyskach nieodwracalnie !

Nazwa wirusa ma swe źródło w tym, że po raz pierwszy został on wykryty i opisany na Lehigh University w USA.

Wirus EXE/AIDS

Jest to przedstawiciel dość licznej klasy wirusów napisanych w językach wyższego poziomu (na ogół w Pascalu), podobnie działających i znanych pod wspólną nazwą AIDS lub Pascal.

Nakłada się na programy typu *.EXE, niszcząc ich oryginalną zawartość i nie zmieniając długości ani innych parametrów programu. Przy każdej próbie uruchomienia zniszczonego przez wirusa programu kasowanych jest kolejnych sześć programów typu *.EXE na dysku C: w katalogu głównym lub dowolnym z jego podkatalogów i jeden program *.EXE w głównym katalogu dysku A:. Dodatkowo w podkatalogach katalogu głównego dysku C: poszukuje jednego zbioru *.DBF, ktory następnie jest niszczony przez zapisanie w kilkuset przypadkowych miejscach po jednym bajcie przypadkowej wartości. Niektore zniszczone bazy danych są przez wirusa w specjalny sposób oznaczane.

Zewnętrznie wirus przejawia się tym, że uruchomiony program wyświetla komunikat i powraca do systemu DOS. Treść komunikatu jest losowana z zawartej w wirusie listy.

Wirus 1701/1704

Dokleja się na końcu programów typu *.COM, zwiekszając ich wielkość o 1701 lub 1704 bajty. Po uruchomieniu programu będącego nosicielem, wirus dokleja się do rezydujacego w pamięci operacyjnej systemu DOS. Od tej chwili śledzi każde uruchomienie programu. Jeżeli jest to niezarażony jeszcze program typu *.COM, zaraża go. Wirus zawiera także drugą procedurę, która generuje przypadkową liczbę taktów zegara, po której niszczy w charakterystyczny sposób zawartość ekranu ("spadające literki"), po czym generuje nową liczbę, mniejszą od poprzedniej itd, aż do momentu, w którym na skutek małego odstępu czasowego między kolejnym zaktywizowaniem się spadających literek, praca na komputerze jest niemożliwa.

Do ciekawostek należy fakt, że wirus sprawdza, czy komputer jest oryginalnym IBM PC i jeżeli tak, to żadnej destrukcji ekranu nie przeprowadza.

Wersja wirusa o długości 1701 jest starsza i zawiera kilka błędów programowych (między innymi błąd w sprawdzaniu IBM'a), usuniętych w nowszej wersji o długości 1704 bajty.

Wirus JEROZOLIMSKI

Dokleja się na początku i końcu programów typu *.COM (zwiekszając ich wielkość łącznie o 1808+5 bajtów), lub na końcu programów typu *.EXE (zwiększa je jednorazowo o 1808 bajtów). Programy typu *.COM zarażane są jeden raz, natomiast *.EXE (przypuszczalnie na skutek błędu) mogą być zarażone wielokrotnie. Po uruchomieniu programu wirus dokleja się do rezydującego w pamięci operacyjnej systemu DOS, po czym zaraża wszystkie uruchamiane programy. Po upływie 32,398 taktów zegara, o ile rok bieżący jest różny od 1987, uruchamiają specjalną procedurę, która najpierw przesuwa zawartość ekranu począwszy od kolumny 5 wiersza 5 do kolumny 1 wiersza 16, a następnie, po każdym przerwaniu zegarowym, wykonuje pętlę spowalniającą działanie komputera. Jeżeli datą bieżącą jest 13 dzień miesiąca i do tego piątek, a rok jest różny od 1987, każdy uruchamiany program będzie kasowany przed próbą uruchomienia go.

Nazwa wirusa pochodzi stąd, że po raz pierwszy wykryto go na Hebrew University w Jerozolimie. Chodzą słuchy (trudno stwierdzić na ile prawdziwe), że miał on być zemstą Palestyńczyków na Izraelu.

Wirus MUZYKANT/OROPAX

Dokleja się na końcu programów typu *.COM, zwiekszając ich wielkość o losowo zmienną wartość od 2756 do 2807 bajtów. Po uruchomieniu programu-nosiciela wirus dokleja się do rezydującego w pamięci operacyjnej systemu DOS. Od tej chwili każda dotycząca dysku operacja DOS'u inicjuje poszukiwanie na dysku, na którym się ta operacja odbywa, niezarażonego jeszcze programu typu *.COM, a następnie jego zainfekowanie. Dodatkowo, jeżeli aktualna data jest późniejsza niż 01-05-1987 i komputer, na którym działamy, jest kompatybilny z IBM PC AT, instalowana jest procedura odgrywająca co pewien czas jeden z zaprogramowanych utworów.

Wirus VACSINA

Instaluje się w pamięci komputera w czasie pierwszego uruchomienia zarażonego nim programu. Robi to, podobnie jak wirusy 1701/1704, w sposób niewidoczny dla rezydentnych programów monitorujących (np. Flushota). Następnie śledzi aktywność systemu operacyjnego, interesując się głównie uruchamianymi programami. Dzięki temu może każdy z nich poddać obróbce przed uruchomieniem. W wypadku programów typu *.COM najpierw sprawdza, czy nie są już one zarażone starszą wersją tego wirusa (wirus z moich zbiorów jest 16 mutacją). Jeśli tak – leczy je. Zdrowy program jest następnie zarażany nową wersją wirusa. To samo dzieje się również przed instalacją wirusa w pamięci – o ile zainstalowała się tam wcześniej starsza wersja zostaje ona usunięta.

W wypadku zbiorów *.EXE została przyjęta oryginalna metoda infekcji. Po napotkaniu zbioru tego typu (co wynika z jego struktury, a nie nazwy) o wielkości nie przekraczającej 64947 bajtów, wirus przetwarza go na *.COM, używając do tego procedury identycznej z wykorzystywaną do przetworzenia na typ *.COM modułów systemu PC-DOS. Następnie zbiór traktowany jest identycznie jak te, które były typu *.COM oryginalnie. Dzięki temu zbiór *.COM powiększa się o 1339 bajtów, a *.EXE – o 1471 bajtów plus 0 do 15 bajtów wyrównujących wielokość programu do wielokrotności 16.

Wirus YANKEE DOODLE

Ten dość ciekawy wirus znalazł się w Polsce już w miesiąc po swej światowej premierze, która miała miejsce 3 sierpnia 1989 w siedzibie ONZ w Wiedniu. Wirus instaluje się w pamięci komputera i z niej zaraża wszystkie uruchamiane programy – zarówno zbiory *.EXE jak i *.COM. Dodatkowo, (o ile rozmnoży się liczbę razy będącą wielokrotnością 8) około godziny 17.00 odgrywa melodyjkę, która użyczyła swej nazwy wirusowi.

Moim zdaniem wirus jest tego samego autorstwa co VACSINA (wynika to z jego kodu) i można go potraktować jako jego nowszą wersję. Tak więc, pisząc o Yankee-Doodle mam na myśli nie jednego wirusa, ale całą ich linię rozwojową (najwyraźniej autor jest bardzo pracowity). Ponieważ wirusy zawierają w sobie numer wersji, można dokładnie określić ich chronologię (wynika ona również z kodu wirusów, przy śledzeniu go widać wyraźnie, jak autor dopracowuje swoje dzieło). Jak na razie trafiły do mnie wersje 34, 39 i 44.

Wirus napisany jest w sposób trudny do zanalizowania, a większa część jego liczącego 2568/2778/2881 bajtów kodu służy do uniemożliwienia śledzenia go przez *debugger.* Między innymi zawiera w sobie sumy korekcyjne, umożliwiające mu przywrócenie oryginalnych wartości zmienionym (na przykład przez wstawienie *breakpointu*) bajtom wewnątrz swojego kodu (sprawdzane 18 razy w ciągu sekundy). Przed zarażeniem wirus sprawdza, czy program nie jest już zainfekowany jego starszą wersją. Jeśli tak, to najpierw leczy go, a następnie zaraża nowszą wersją. Wirus wykrywa również fakt rezydowania swojej starszej wersji w pamięci komputera. W takim przypadku rezydująca część jest dezaktywowana. Wirus monitoruje również odwołania do DOS'u i przy pewnej kombinacji wartości rejestrów zmienia swój stan na taki, w którym zamiast zarażać, leczy programy przez siebie zainfekowane.

VACSINA jest traktowany na równi ze starszymi wersjami Yankee-Doodle'a.

Wirus DARK AVENGER

Pomimo angielskiej nazwy "MROCZNY MŚCICIEL" powstał w Bułgarii. Należy do grupy wirusów zarażających programy. Po uruchomieniu nosiciela instaluje się w pamięci komputera i stamtąd zaraża wszystkie widziane zbiory z rozszerzeniem .COM i .EXE (na których dokonywane są jakieś operacje) oraz wszystkie zbiory, które zostają uruchomione – niezależnie od ich rozszerzenia (mogą być to OVL, . BN, . CHN, . DRV i inne, zależnie od inwencji programisty piszącego program). Dodatkowo, po każdym piętnastym uruchomieniu dowolnego zarażonego programu (licznik znajduje się w boot sektorze), wirus losuje jeden z sektorów powyżej głównego katalogu na aktualnym dysku i zamazuje go niesłychanie dowcipną informacją, zaczynającą się od tekstu:
"Eddie lives...somewhere in time!".

Trafić może takim strzałem w dowolne miejsce – jak masz szczęście to w wolny obszar, jak nie – to w środek jakiegoś zbioru lub podkatalog (nietrudno zgadnąć, jaki będą tego konsekwencję. Wirus napisany jest z dużym znawstwem – na przykład odwołania do dysku odbywają się przez bezpośredni skok do pamięci ROM BIOS'u w sposób całkowicie niewidzialny dla programów zabezpieczających w rodzaju Flushota.
DARK AVENGER TO WIRUS BARDZO NIEBEZPIECZNY!!!

Wirus WŁOSKI

Jest przedstawicielem rodziny wirusów, które nie zarażają programów, lecz całe dyski. Rezyduje na dysku w obszarze *BOOT* sektora oraz jednokilobajtowym obszarze, oznaczonym w tablicy FAT jako miejsce o uszkodzonym nośniku. Aktywizacja wirusa następuje w momencie ładowania systemu operacyjnego z zarażonego dysku. Wirus instaluje się w pamięci komputera zanim zostanie załadowany system i przejmuje obsługę operacji dyskowych. Od tego momentu każda operacja wykonana na dysku (wystarczy dotycząca go dyrektywa DIR, czy nawet uczynienie go dyskiem aktualnym) powoduje jego zarażenie niezależnie od tego, czy jest to dysk twardy, czy dyskietka. Dodatkowo, podczas każdej operacji odczytu z dysku, sprawdzany jest stan *timera* systemowego. Gdy spełnione są pewne dodatkowe warunki aktywizowana jest procedura inicjująca efekt odbijającej się na ekranie piłeczki (współbieżnie z normalną pracą).

Ostatnio pojawiła się jego kolejna wersja, wyposażona w licznik generacji. Egzemplarz wirusa znajdujący się w moich zbiorach przeszedł przez 26 komputerów. Świadczy to o jego dużej żywotności.

Wirus PAKISTAŃSKI albo BRAIN

Również należy do wirusów zarażających dyski. Potrafi zainfekować wyłącznie dyskietki formatu 360 KB. Dyskietkom innych formatów oraz dyskom twardym nic z jego strony nie grozi. Na zarażonej dyskietce rezyduje w obszarze *BOOT*-sektora oraz w 6 sektorach dysku oznaczonych w tablicy FAT jako miejsca o uszkodzonym nośniku. W głównym katalogu wirus umieszcza swój podpis: nadaje dyskietce nazwę " (c) Brain ".

Jego uruchomienie następuje w momencie ładowania systemu operacyjnego z zarażonej dyskietki. Wirus umieszcza swój kod w siedmiokilobajtowym, ukrytym przed DOS-em obszarze pamięci komputera i stamtąd reprodukuje się na wszystkie czytane przez komputer dyskietki. Poza tym zawiera pewien mechanizm samoobrony, dostarczający programowi czytającemu BOOT-sektor jego oryginalną, a nie aktualną (będącą kodem wirusa) zawartość. Utrudnia to pisanie programów wykrywających go. Wirus nie prowadzi żadnej działalności destrukcyjnej.

Kod wirusa zawiera zabawną dedykację: *"Dedykowany pamięci milionów wirusów, których już nie ma wśród nas".*

Należy do grupy mało szkodliwych i trudno rozpowszechniających się. Fakt jego szybkiej ekspansji w Polsce świadczy o dużej beztrosce użytkowników komputerów. Brain posiada liczne wersje.

Wirus DEN ZUK

Należy do grupy zarażającej wyłącznie dyskietki. Kod wirusa umieszczany jest w specjalny, trudny do złamania sposób na dodatkowej, nie widzianej przez DOS ścieżce. W czasie ładowania systemu DOS z zarażonej dyskietki wirus zagnieżdża się w pamięci komputera, przechwytując obsługę dysków w sposób umożliwia-

> 32

31 <

jący mu zarażanie kolejnych dyskietek. Czyni to niewidocznie dla różnych programów antywirusowych, bazujących na blokowaniu niepowołanego dostępu do dysku (na przykład Flushota). Objawia się na zarażonym komputerze (o ile wyposażony jest on w inną kartę monitora niż Hercules) charakterystycznym znakiem firmowym w momencie naciśnięcia kombinacji klawiszy Alt+Ctrl+Del (polecenie przeładowania systemu). Na zarażonym komputerze również kombinacja klawiszy Alt+Ctrl+F5 powoduje przeładowanie systemu.

Po wykonaniu DISKCOPY zarażonej dyskietki systemowej nie można z niej załadować systemu – pojawia się komunikat THE '3 WISE MEN'....
Jnsert DOS diske tte and REBOOT when ready

Przed zarażeniem dyskietki wirus sprawdza, czy nie jest ona już zainfekowana wirusem Brain, a jeżeli tak – likwiduje go.

Wirus STONED

Nie zaraża programów, lecz całe dyski lub dyskietki. Zarażany jest pierwszy (fizycznie) dysk twardy, niezależnie od tego, czy jest na nim partycja DOS-u. Wirus lokuje się w tablicy partycji dysku oraz na zerowej ścieżce bezpośrednio za nią. W przypadku dysku w całości przeznaczonego na DOS, jest to obszar wolny, natomiast na dyskach z partycjami Novell Netware, UNIX lub XENIX mogą ulec zamazaniu informacje tych systemów. Wirus instaluje się również na wszystkich dyskietkach włożonych do napędu A, do których system choć raz się odwoła. Na dyskietkach wirus umieszcza swój kod w *boot* sektorze DOS-u oraz w jednym z sektorów z głównego katalogu dyskietki. Na dyskietkach 360 Kb jest to ostatni, 7 sektor. Dla dyskietek 720 Kb i 1.2 Mb są to odpowiednio 5 i 3 sektor tego katalogu. Wirus nie sprawdza, czy sektory te są wolne. W przypadku wielu zbiorów na dyskietce opisy ostatnich mogą ulec zniszczeniu, co oznacza ich utratę. Prawdopodobieństwo takiego zdarzenia rośnie wraz z pojemnością dyskietki. Ewentualna destrukcja w czasie rozmnażania się – to jedyny czynnik niszczący tego wirusa.

Wirus dekonspiruje się czasem podczas ładowania systemuu DOS z zarażonej dyskietki wyświetlając komunikat: *"Your PC is now Stoned!"* lub *"Twój PC jest teraz be!"* w spolszczonej przez rodzimego "terrorystę" wersji.

Wirus YALE/ALAMEDA

Zaraża wyłącznie dyskietki znajdujące się w napędzie A: w momencie naciśnięcia klawiszy Ctrl-Alt-Del (przeładowanie systemu) na zarażonym komputerze. Infekcja komputera następuje w wyniku załadowania systemu DOS z zarażonej dyskietki. Wirus umiejscawia się w *boot* sektorze dyskietki, przepisując jego oryginalną zawartość do sektora 8 zerowej strony 39 ścieżki. Nie sprawdza, czy sektor ten jest wolny, może więc zamazać informację w zbiorze, który się tam znajduje. Po zapisie status sektora nie ulega zmianie. Może to prowadzić do jego zapisania już po infekcji. Skutkiem będzie zawieszanie się komputera przy próbie ładowania systemu.

Kod wirusa zawiera teoretycznie nielegalną, lecz na procesorach 8088 i 8086 działająca instrukcję:

POP CS

W przypadku procesora 80286 i następnych jest ona już naprawdę nielegalna. Tak więc, wirus ten może zarazić wyłącznie komputery PC XT. Na PC AT próba ładowania systemu z zarażonej dyskietki zawsze doprowadzi do zawieszenie się komputera.

Istnieje już wersja wirusa (nie wpadła mi dotąd w ręce) pozbawiona powyższej "niedogodności".

Wirus Disk Killer/Ogre

To mój najnowszy "nabytek". Dotarł do Polski na firmowej dyskietce zamkniętej w zaplombowanej kopercie, zawierającej oprogramowanie towarzyszące myszy ADVANCE. Mysz pochodziła z partii kilkuset sztuk, która świeżo dotarła do Warszawy. Jest prawie pewne, że wirus jest na wszystkich dyskietkach tej dostawy.

Wirus umiejscawia się w *boot* sektorze oraz w dwóch *clusterach* dysku, oznaczonych jako obszar o uszkodzonym nośniku. Uruchomienie następuje w momencie ładowania systemu z zarażonej dyskietki lub omyłkowego naciśniecia sekwencji klawiszy Ctrl-Alt-Del w momencie, gdy zarażona dyskietka (niekoniecznie z systemem operacyjnym) znajduje się w napędzie A:. Wirus instaluje się w pamięci i stamtąd mnoży się na kolejne dyski. Jednocześnie prowadzi licznik czasu pracy komputera z wirusem w pamięci. Gdy sumaryczny czas działania osiągnie 48 godzin od chwili infekcji, wirus wyświetla komunikat:

Disk Killer -- Version 1.00 by COMPUTER OGRE 04/01/1989
Warning!!
Don't turn off the power or remove the diskette while Disk Killer is Processing!
PROCESSING

i rozpoczyna destrukcję, sprowadzającą się do zaszyfrowania całej zawartej na dysku twardym informacji. O zakończeniu pracy melduje komunikatem:

Now you can turn off the power.
I wish you luck !

Wbrew rozpowszechnianym na Zachodzie informacjom dysk nie jest niszczony – operacja szyfrowania jest odwracalna – o ile wczesniej nie został przerwany proces szyfrowania.

Pakiet programow antywirusowych rozpowszechniany przez redakcję PMI "KOMPUTER" pomoże Ci w walce z następującymi wirusami:

Program:

5FSP_170
SCANVIR
CLEANVIR
AZOTOX
CHKUP
NOVIRUS
VIRUS
CHK4BOMB
PROGNOSE
MKS DEMO

Program:	MKS DEMO	PROGNOSE	CHK4BOMB	VIRUS	NOVIRUS	CHKUP	AZOTOX	CLEANVIR	SCANVIR	5FSP_170
Wirus:	**V**	**V**	**V**	**V**	**V**	**V**	**V**	**V**	**V**	**V**
507	W	.	.	W	W	W	.	.	.	W
534	U	.	.	W	W	W	.	.	.	W
648	U	.	.	W	W	W	U	.	.	W
CHOINKA	W	.	.	W	W	W	.	.	.	.
LEHIGH	U	.	.	.	.	.	.	U	W	.
EXE/AIDS	U	.	.	.	.	.	.	U	W	W
1701	U	.	.	W	W	W	.	U	W	.
1704	U	.	.	W	W	W	.	U	W	.
JEROZOLIMSKI	U	.	.	W	W	W	.	U	W	W
MUZYKANT/OROPAX	U	.	.	W	W	W	.	U	W	W
VACSINA	W	.	.	W	W	W	.	U	W	.
YANKEE DOODLE	W	.	.	W	W	W	.	U	W	.
DARK AVENGER	W	.	.	W	W	W	.	U	W	.
WŁOSKI	U	.	.	W	W	W	.	U	W	W
PAKISTAŃSKI albo BRAIN	U	.	.	W	W	W	.	U	W	W
DEN ZUK	W	.	.	.	.	.	.	U	W	.
STONED	U	.	.	.	.	.	.	U	W	W
YALE/ALAMEDA	U	.	.	.	.	.	.	U	W	W
Disk Killer/Ogre	W	.	.	.	.	.	.	U	W	.

Legenda:
W – tylko wykrywa **U – wykrywa i usuwa**

FSP_170 – program jest nieskuteczny wobec wirusów bootsektora oraz wobec Yankee-Doodle-a i Dark Avengera. Pozostałe z wymienionych tu wirusów powinny zostać przezeń wychwycone.

SCANVIR – zawiera dość długą listę wykrywanych wirusów. Jest jednak produktem amerykańskim i nie wykrywa wirusów rozpowszecnionych głównie w Europie, m.in. takich jak Choinka, 507, EXE/AIDS.

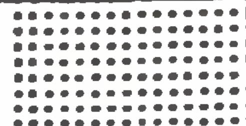
CLEANVIR – pakiet stowarzyszony z pakietem SCANVIR – usuwa wirusy ze wspomnianej listy programu SCANVIR.

AZOTOX – wykrywa i niszczy jednego tylko wirusa – 648.

CHKUP, NOVIRUS, VIRUS – pakiety wykrywające wirusy przez porównanie obliczonych sum kontrolnych z tymi, które wcześniej zostały zapisane w bazie.

CHK4BOMB, PROGNOSE – pakiety analizujące kod programu i oceniające, czy są w nim jakieś niebezpieczne instrukcje. Moim zdaniem mają one niewielkie praktyczne zastosowanie.

MKS_VIR – wykrywa wszystkie opisane tu wirusy. Wszystkie wykryte wirusy usuwane są przez wersję licencjonowaną programu. Zawarta w FIDO wersja DEMO kilku nie usuwa.

  \W pracy\ 

Zenon Rudak

Komputer z mikrometrem



Gdy konstruktor narysuje na desce kreślarskiej elementy nowej konstrukcji, do pracy zabiera się technolog, a następnie dział budowy prototypów. I tak powstaje produkt finalny – nowy wyrób. Prototyp wykonuje się z materiałów, jakie będą używane przy jego seryjnej produkcji, lub z materiałów zastępczych, pozwalających na ocenę kształtu czy funkcji. Do przeniesienia kształtu nowego przedmiotu z rysunku konstrukcyjnego na materiał używa się znanych wszystkim narzędzi jak: linijki, przymiary, wzorniki, stoły pomiarowe z elementami bazowymi itp. Wykonany prototyp poddaje się badaniom, oględzinom, wtórnej analizie konstrukcyjnej w celu optymalizacji zakładanych funkcji i maksymalnego uproszczenia późniejszej produkcji. Wynikiem optymalizacji jest wykonanie narzędzi specjalnych służących do produkcji nowego wyrobu np. forma odlewnicza, tłocznik do blachy, kształtowy nóż lub frez skrawający itp. Cały proces tworzenia prototypu i dalszej pracy nad uruchomieniem produkcji nierozerwalnie związany jest z procesami pomiaru odległości.

Jeżeli nowy przedmiot jest stosunkowo mały, a na jego kształt składa się wiele przecinających się płaszczyzn, to pomiary i ustalenie położenia punktów jego powierzchni w przestrzeni jest dość proste (np. pudełko do kasety magnetofonowej). Jeżeli natomiast przedmiot jest duży i ma skomplikowany kształt (np. śruba okrętowa, elementy karoserii samochodu), ustalenie położenia punktów jego powierzchni jest bardzo trudne i wymaga stosowania specjalnych metod pomiarowych. Polegają one na tworzeniu przestrzennego układu współrzędnych z wpisanymi liniami siatki współrzędnych. Niejednokrotnie rolę siatki odniesienia pełnią rozpięte i naprężone nici lub wiązki promieni światła. Wśród takiej siatki umieszcza się poszczególne elementy budowanego prototypu. Siatka współrzędnych pozwala na poprawne ustalenie położenia poszczególnych elementów i na przeprowadzenie pomiarów niezbędnych do dalszej pracy nad narzędziami do produkcji konstruowanej części. Do zachowania należytej jakości późniejszej produkcji wszystkie pomiary muszą być przeprowadzone z jak największą dokładnością. W praktyce pomiary elementów takich jak karoserie samochodów, obudowy maszyn, części silników wykonuje się na specjalnych żeliwnych stołach z naciętą na ich powierzchni siatką pomiarową (płaszczyzna x-y). Sposób wykonania stołu pomiarowego musi gwarantować jednakowy błąd pomiaru długości, bez względu na miejsce położenia przedmiotu mierzonego i jego wymiary. Siatka stołu służy do orientowania elementów pomiarowych przeznaczonych do mierzenia odległości trzeciej współrzędnej – wysokości. W ten sposób przeprowadza się pomiary w wielu zakładach przemysłowych do dziś. Postęp techniczny umożliwił wykonywanie bardzo dokładnych stołów pomiarowych i używanie precyzyjnych optycznych narzędzi pomiarowych.

Obecnie nadszedł czas na zastosowanie nowoczesnej techniki komputerowej, która pozwala na zautomatyzowanie i znaczne ułatwienie skomplikowanych pomiarów w przestrzeni. Pozwala także na rezygnację z bardzo drogich i trudnych do wykonania stołów pomiarowych.

Jako przykład nowoczesnego urządzenia pomiarowego przedstawię komputerowy system pomiaru odległości produkowany przez zachodnioniemiecką firmę **Opton**.

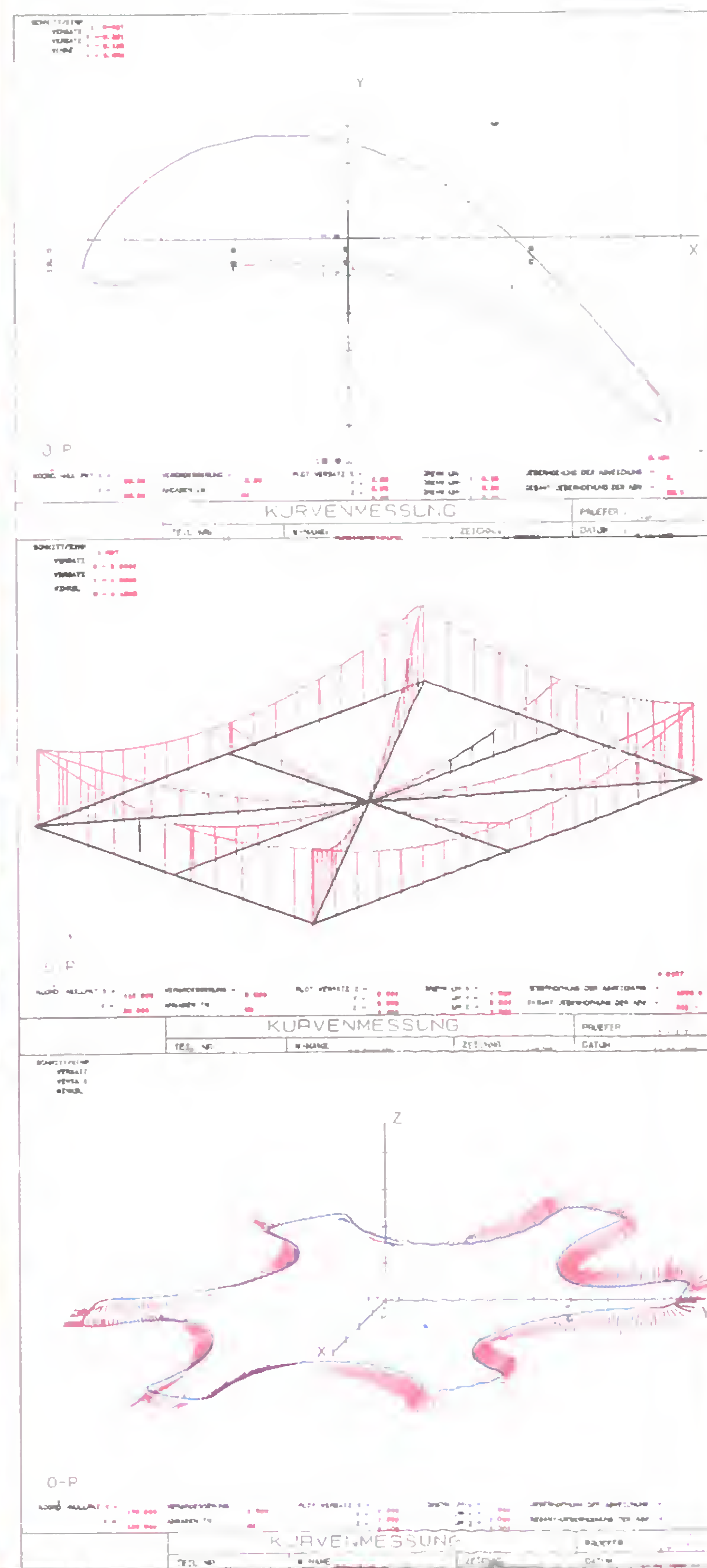
System składa się: z komputera klasy IBM PC/AT służącego do wprowadzania danych o badanym przedmiocie, systemu sterowników wieloprocesorowych przeznaczonych do obsługi, poruszania i kontroli położenia "ramienia" czujnika pomiarowego, jednostki centralnej odbierającej i przetwarzającej sygnały z czujnika pomiarowego, zespołu pamięci masowej do zbierania danych dostarcza-



> 34







33 <

nych z czujnika oraz peryferyjnych urządzeń drukujących, pozwalających na liczbowo-graficzne odczytywanie wyników pomiarów.

Praca systemu pomiarowego rozpoczyna się od wprowadzenia do pamięci komputera danych o powstającej konstrukcji. Dane te mogą być przygotowane w dwojaki sposób. Mogą to być liczby opisujące współrzędne przestrzenne lub dane w postaci rysunku. Dane liczbowe powinny być opracowane przez konstruktora i muszą odpowiadać przyjętym jednostkom pomiarowym. Lepsza i ciekawsza jest forma wprowadzania danych w postaci rysunku. Służy do tego program rodziny CAD. Twórca rysuje swą konstrukcję na ekranie komputera za pomocą digitizera. Konstruowanie odbywa się w trójwymiarowym układzie współrzędnych. Powstające rysunki są przez program automatycznie "tłumaczone" na komendy sterujące systemem pomiarowym. Komputerowy rysunek konstrukcji stanowi jednocześnie dokumentację niezbędną dla pracowników budujących prototyp.

Dalszym etapem pracy systemu jest umieszczenie na stole pomiarowym części przeznaczonej do zwymiarowania. Do mierzonych elementów dosuwane jest ramię z czujnikiem pomiarowym. Końcówka czujnika może poruszać się w trójwymiarowej przestrzeni. Ramię czujnika poruszane jest przez operatora za pomocą układu zdalnego sterowania lub przez komputer na podstawie określonego wcześniej programu. Program ruchu ramienia powstaje w wyniku zapamiętania w pamięci układu sterującego poleceń wydawanych przez operatora przy pierwszym cyklu pomiarowym. System taki pozwala na całkowitą i bezbłędną powtarzalność cyklu pomiarowego i wybranych mierzonych miejsc. Sterowaniem ramienia czujnika pomiarowego zajmuje się procesorowy układ elektroniczny, zarządzający hydraulicznymi elementami wykonawczymi złożonymi z popychaczy, pomp, siłowników. Ramię pomiarowe zakończone jest czujnikiem. Czujnik może być mechaniczny, optyczny lub pneumatyczny. Dostarcza on informacji o położeniu w przestrzeni wskazanych programem lub przez operatora punktów pomiarowych. Dane o położeniu punktów pomiarowych magazynowane są w pamięci masowej, a po zakończeniu pomiarów analizowane przez jednostkę centralną. Analiza polega na wykazaniu odchyleń pomiaru rzeczywistego od wielkości, jakie podane są w projekcie. Wynik analizy przedstawiany jest w dwóch formach: tabelarycznej – kolumny cyfr określające położenia poszczególnych punktów pomiarowych i ich odchylenia w stosunku do projektu, lub w formie graficznej – nałożone na siebie dwa rysunki. Jeden w kolorze czarnym stanowiący wzorzec podany przez konstruktora, drugi w kolorze czerwonym odzwierciedlający rzeczywisty kształt badanej części.

Opisany system zapewnia uzyskanie dokładności pomiarowej wynoszącej 0,001 mm dla elementów nie przekraczających 2,5 m długości. Rozbudowane, opisane tu systemy mogą obsługiwać elementy o długości do 20 m, szerokości do 4,5 m i wysokości do 2,5 m.

Przedstawiony system pomiarowy może być wykorzystywany także do pracy w "odwrotnym" kierunku. Operator obsługujący urządzenie może wybierać punkty pomiarowe badanego elementu i dostarczać dane opisujące jego kształt. Zebrane w ten sposób dane mogą posłużyć do stworzenia dokumentacji technicznej badanej części. Zależnie od ilości zebranych informacji o pomiarach, dokumentacja może powstać w wyniku działania programu analizującego lub w wyniku ręcznego łączenia określonych przy pomiarze punktów.

Innym zastosowaniem takich systemów jest sterowanie frezarko-kopiarkami lub automatami tokarskimi. System dostarcza stałych danych o kształcie mierzonego wzorca, a sprzężona z systemem frezarko-kopiarka odwzorowuje ten kształt na innym materiale.

Kolejne zastosowanie to kontrola bieżącej produkcji. System na podstawie zapisanego w pamięci programu kontroluje wybrane elementy. Zaletą tego zastosowania jest całkowita powtarzalność pomiarów oraz statystyczne opracowywanie wyników dla każdej mierzonej partii wyrobów.







Podstawowe dane techniczne komputera Mac IIfx:	
Procesor:	Motorola 68030
Częstotliwość zegara:	40 MHz
Koprocesor:	Motorola 68882
Pamięć RAM:	4 MB z możliwością rozszerzenia do 32 MB
Pamięć ROM:	512 KB
Pamięć obrazu:	2 MB
Pamięć zewnętrzna:	stacja dyskietek 3,5 cala 1,44 MB dysk twardy 80 MB (160 MB)
Interfejsy:	dwa RS 232, Apple Talk
Monitor:	kolor 14 cali
Rozdzielczość ekranu:	640 na 480 punktów Minimalna średnica punktu na ekranie: 0,21 mm

W ubiegłym roku pisaliśmy o komputerach Apple Macintosh kilkakrotnie. Każda wzmianka podkreślała nowoczesność i ogromną szybkość pracy tych maszyn. Tym razem też będzie o komputerze z rodziny kolorowego jabłuszka, który można bez wahania określić mianem sprintera rekordzisty. Mowa tu o komputerze Mac IIfx najnowszej konstrukcji firmy Apple.

Maszyna IIfx na pozór nie różni się od swoich poprzedników. Obudowa ta sama co w pierwszym modelu Mac II, podobna klawiatura i myszka, ta sama szata graficzna i kolorystyczna. Wszystko niby to samo, ale...i tu zaczynają się różnice. Komputer ma nową płytę główną z sześcioma złączami typu NuBus dla kart rozszerzenia i jednym złączem dostępu bezpośredniego typu PDS (Processor Direct Slot). Złącza typu NuBus miał każdy poprzednik, a złącze PDS po raz pierwszy zastosowano w młodszym bracie – modelu Mac SE/30.

Nowy Mac to konstrukcja w pełni wieloprocesorowa. Nie ma tu żadnej przesady ani dowcipnego skojarzenia. Konstruktorzy postawili sobie za cel osiągnięcie jak nawiększej szybkości pracy komputera. By to uzyskać wszelkie operacje pomocnicze powierzono mikroprocesorom odciążającym procesor główny. Bezpośrednim dostępem do pamięci (DMA) steruje kontroler ze specjalizowanym procesorem. Sterownik typu SCSI będący specjalizowanym układem mikroprocesorowym obsługuje pamięć masową. Porty wejścia wyjścia (Apple Desktop Bus i interfejsów RS 232) mają dwa sterowniki z procesorami 8- bitowymi 6502, taktowane zegarem 10 MHz. Wyliczając użyte procesory należy wspomnieć, że procesor główny komputera to Motorola 68030 taktowany zegarem 40 MHz i współpracujący bezpośrednio z koprocesorem matematycznym 68882. Dla przyśpieszenia komunikacji procesora z pamięcią operacyjną i masową przyłączono do procesora układ pamięci podręcznej (cache) o pojemności 32 KB składającej się ze statycznych układów RAM o czasie dostępu 25 nanosekund.

Co daje tak skomplikowana i bogata w procesory konstrukcja? Znacznie przyśpiesza działanie komputera. Procesory pomocnicze odciążają procesor główny od oczekiwania na wykonanie nakazanych działań przez inne elementy komputera. Zajmuje się on wtedy wykonywaniem programu bez konieczności sterowania wszystkimi interfejsami i odczytywania stanu portów wejścia wyjścia. Taka konstrukcja pozwala na około trzykrotne przyśpieszenie pracy w stosunku do poprzedniego modelu SE/30, uważanego za najszybszy komputer klasy PC (Personal Computer). Mac IIfx wykonuje operacje komunikacji z dyskiem twardym i pamięcią operacyjną prawie siedem razy szybciej, a operacje przewijania ekranu, przeszukiwania rekordów baz danych i rysowania elementów graficznych na ekranie dwa razy szybciej niż model SE/30.



W konstrukcji modelu fx użyto pamięci RAM i ROM zorganizowane w postaci zintegrowanych pakietów typu SIMM. Standardowo Mac fx wyposażony jest w cztery pakiety RAM o pojemności 1 MB każdy i jeden pakiet ROM o pojemności 512 KB. W pamięci ROM zapisano nowy system operacyjny w wersji 7.0. Zastosowanie procesorów we wszystkich interfejsach komputera spowodowało zmiany w systemie operacyjnym. Pamięć RAM można rozszerzyć do 8 MB (stosując kolejne pakiety SIMM) lub do 32 MB (używając ich wersji o pojemności 4 MB). Użyte w fx pakiety RAM posiadają 64 stykowe połączenia. W poprzednich modelach miały złącza 32 stykowe. Taka filozofia konstrukcji znacznie utrudnia rozbudowę posiadanego sprzętu.

Pamięć masowa komputera fx to wbudowana stacja dyskietek 3,5 calowych i dysk twardy 80 lub 160 MB.

Komfort pracy operatora podniesiono, zmieniając konstrukcję wentylatora chłodzącego wnętrze komputera. Nowy wentylator ma zwiększoną średnicę z 80 do 92 milimetrów, co pozwoliło zmniejszyć prędkość obrotową o około 30 procent z zachowaniem tej samej wydajności chłodzenia. Obniżyło to jego hałaśliwość o ok.4,5 dB. Dodatkowo specjalny układ steruje wydajnością chłodzenia przez zmiany szybkości obrotowej wentylatora zależnie od obciążenia cieplnego komputera.

Zmiany konstrukcyjne nie objęły interfejsu Apple Talk (służącego do przesyłania danych w sieci i komunikacji z drukarką) oraz układu generowania dźwięku.

Na uwagę zasługuje nowa karta graficzna o nazwie 8/24 GC. Wyświetla ona kolorowy obraz z 8 – lub 24 – bitowym opisem jednego punktu na ekranie z rozdzielczością 640 na 480 pikseli. Maksymalnie karta tworzy paletę o 16 milionach kolorów. Pracuje również w trybie mono z 1-, 2-, 4- lub 8- bitowym opisem każdego punktu ekranu. Rozdzielczość obrazu w tym przypadku wynosi 1152 na 870 punktów a maksymalna skala szarości obejmuje 256 stopni. Sercem karty 8/24 jest procesor AMD29000 typu RISC taktowany zegarem 30 MHz. Procesor współpracuje z pamięcią obrazu o pojemności 2 MB. Można ją rozbudować do 4 MB. Używa się do tego tych samych pakietów SIMM, którymi rozszerza się pamięć operacyjną komputera. Aby wyświetlanie obrazu dopasować do szybkości pracy nowej maszyny zastosowano w karcie graficznej 64 KB pamięci podręcznej, ułatwiającej procesorowi przesyłanie danych o poszczególnych punktach obrazu. Nowa karta graficzna pracuje około 30 razy szybciej, niż jej poprzedniczki używane w starszych modelach Maca.

Mac IIfx kosztuje w wersji podstawowej około 12000 USD. Za nową kartę graficzną trzeba dopłacić 2100 USD.

Mac IIfx należy do najszybszych komputerów klasy PC na świecie. Maszyna ta polecana jest użytkownikom zajmującym się skomplikowanymi obliczeniami inżynierskimi, przetwarzaniem i animacją grafiki, pracą z programami kontrolowanymi przez system UNIX i pracą w sieciach komputerowych.

Nowy Mac IIfx zadziwia swą olbrzymią szybkością pracy, szczególnie gdy obserwuje się przetwarzanie animowanej kolorowej grafiki. czy analizę trójwymiarowych konstrukcji tworzonych w programach typu CAD.

I pomyśleć, że gdyby dzieci rosły tak szybko, jak rozwija się technika procesorów... Może to jednak lepiej, że tylko komputery są coraz szybsze.

\prawo autorskie\

Andrzej Popławski

OCHRONA WŁASNOŚCI INTELEKTUALNEJ

Najwyraźniej w Europie Wschodniej coś się ruszyło w dziedzinie ochrony praw autorskich na oprogramowanie. Mówi się o tym ostatnio coraz więcej: władze radzieckie zobowiązały się do uregulowania tych spraw w najbliższym czasie, Polska również przygotowuje się do wprowadzenia niezbędnych zmian prawnych. Jest to warunek normalnego handlu programami w skali nie tylko międzynarodowej, ale i wewnętrznej. W najbliższym czasie będzie zorganizowana międzynarodowa impreza, która może stać się istotnym bodźcem dla wprowadzania nowych rozwiązań legislacyjnych, chroniących prawa autorów oprogramowania. W dniach 18-20 lutego 1991 w Pradze odbędzie się konferencja "Własność intelektualna w dziedzinie oprogramowania" (Intellectual Property in Software). W konferencji wezmą udział przedstawiciele wielu firm i organizacji z krajów Europy Wschodniej i Zachodniej, zajmujących się prawną stroną ochrony praw autorów programów.

Nowa sytuacja polityczna i gospodarcza w Europie (nie tylko Wschodniej, bo trzeba pamiętać o niedawnym połączeniu Niemiec i zbliżającym się zjednoczeniu Europy Zachodniej) stwarza absolutnie nowe możliwości w dziedzinie handlu międzynarodowego i współpracy gospodarczej. Jedną z najbardziej interesujących dziedzin wymiany jest niewątpliwie oprogramowanie komercyjne. Jednakże uruchomienie przepływu programów jako towaru możliwe jest dopiero wówczas, gdy wszystkie strony takiej wymiany będą miały pełną świadomość warunków prawnych handlu programami, zarówno w krajach, w których są one wytwarzane, jak i w tych, gdzie mają być sprzedawane. Wiadomo, że Europa Wschodnia stanowi potencjalnie ogromny rynek. Jednakże partnerzy zachodni chcą wiedzieć, co się dzieje w tych krajach w dziedzinie ochrony praw autorskich.

Z kolei przedstawiciele krajów postkomunistycznych powinni dobrze zorientować się w mechanizmach ochrony oprogramowania w krajach zachodnich. Dałoby to możliwość uświadomienia luk w istniejącym systemie prawnym i zastanowienia się nad możliwościami wykorzystania funkcjonujących na Zachodzie rozwiązań. Tym wszystkim zagadnieniom ma być poświęcona praska konferencja.

Pierwsze dwa dni konferencji poświęcone będą głównie wykładom, których podstawowym tematem będzie właśnie ochrona prawna oprogramowania w różnych krajach oraz jej wpływ na handel między Wschodem i Zachodem. Również interesującym dla niektórych specjalistów może być zagadnienie ochrony prawnej konstrukcji układów scalonych. Trzeci dzień zaplanowany jest w sposób elastyczny: zależnie od potrzeb uczestników zostaną zorganizowane seminaria i dyskusje dotyczące rozmaitych tematów szczegółowych. Również tego dnia uczestnicy będą mogli uzyskać interesujące ich informacje w sprawach ekonomicznych, które, być może, pozwolą na nawiązanie nowych kontaktów między wshodnimi i zachodnimi partnerami. Kilka czołowych firm produkujących oprogramowanie przedstawi swe najnowsze opracowania.

Konferencja organizowana jest przez Instytut praw autorskich, ochrony własności przemysłowej i prawnych aspektów konkurencji praskiego Uniwersytetu Karola we współpracy z Ośrodkiem badań prawa handlowego Uniwersytetu Londyńskiego (The Centre for Commercial Law Studies at the Queen Mary and Westfield College, Uniwersity of London) oraz czechosłowacką agencją ochrony praw autorskich AURA-PONT, praską firmą NUMEN-2N i londyńską firmą Dareheath Ltd. Sponsorem konferencji jest londyńska firma wydawnicza Fourjays Publications, wydająca w Wielkiej Brytanii czasopismo "Managing Intellectual Property". Od sponsora uczestnicy konferencji otrzymają bezpłatną roczną prenumeratę tego czasopisma (normalnie roczna prenumerata kosztuje 200 dolarów USA). Dla delegatów z Bułgarii, Czechosłowacji, Jugosławii, Polski, Rumunii, Węgier, i ZSRR opłata za uczestnictwo w konferencji wynosi tylko 200 dolarów USA. Opłata może być również wniesiona w koronach czechosłowackich, według kursu w dniu dokonania przelewu.

Wszelkie informacje oraz blankiety rejestracyjne dla uczestników można uzyskać pod następującym adresem:

Administration, NUMEN-2N
Na Nivach 18 140 00 Praha 4
Czechosłowacja
tel. (422) 42 40 28, fax (422) 42 58 66

Intersoftland

00-519 Warszawa, ul. Wspólna 41 m 49
TEL. 29-59-77

POLECAMY DUŻY WYBÓR DOKUMENTACJI W JĘZYKU POLSKIM DO KOMPUTERÓW IBM:

A. JĘZYKI PROGRAMOWANIA, BIBLIOTEKI
1. Programowanie w Assemblerze (R. Wacławek)
2. Programowanie w GW-Basic (R. Wacławek)
3. GW-Basic - kompilator
4. Programowanie w Turbo-Basic (R. Wacławek)
5. Turbo-Basic v. 1.0 - 1.1
6. Turbo Assembler, Turbo Debugger
7. Turbo Pascal v 4.0
8. Programowanie w Turbo Pascal v. 5.0 (R. Wacławek)
9. Turbo Pascal v. 5.0
10. Turbo Pascal v. 5.5 (dodatek do v. 5.0)
11. Turbo Database Toolbox do TP3
12. Turbo Database Toolbox do TP4/5
13. Turbo Power Tools do TP4/5
14. Metody numeryczne do TP4/5
15. Turbo Prolog v. 1.0
16. Programowanie w Turbo C v. 2.0 (R. Wacławek)
17. Turbo C v. 1.0
18. Turbo C v. 1.5
19. Turbo C v. 2.0
20. Aztec C v. 3.2
21. Zastosowanie języka C dla zaawansowanych
22. Quick C
23. Język C w systemie UNIX
24. Programowanie w języku Fortran 77
25. Agraph, bibl. do języków Fortran i Pascal
26. SSP/PC, procedury numeryczne do Fortranu
27. Modula 2 Logitech

B. BAZY DANYCH, PAKIETY ZINTEGROWANE, PROGRAMY KALKULACYJNE
1. Przewodnik programisty po dBase II/III
2. dBase III, poradnik encyklopedyczny
3. POLONUS, kompletny opis + całkowicie spolszczony program dBase III+ (R. Wacławek)
- 4a. dBase III+, programowanie
- 4b. dBase III+, poznawanie
- 4c. dBase III+, zastosowania
- 4d. dBase III+, praca w sieci
- 4e. dBase III+, instalacja
- 4f. dBase III+, generator aplikacji
5. dBase III+, opis komend
6. Co nowego w dBase IV
7. dBase IV, rozkazy i funkcje
- 8a. dBase IV, opis języka
- 8b. dBase IV, zarządzanie bazą danych
- 8c. dBase IV, programowanie, język SQL, praca w sieci
- 8d. dBase IV, przewodnik po menu
9. Clipper SARMATA 87, pełny opis + całkowicie spolszczony program (R. Wacławek)
10. Sarmagra, pakiet graficzny do SARMATY, opis + program (R. Wacławek)
11. Clipper 87, opis
12. Clipper 87, compendium
13. Clipper 87, biblioteka IDL.LIB
14. Praktyka programowania w Clipperze
15. PLIB, PLINK 86
16. Fox Base +
17. Informix v. 3.2 (DOS)
18. C-ISAM, pakiet procedur do Informixa
19. Lotus 1-2-3 v. 2.0
20. Grafika 3D do Lotusa i Symphony
21. Stat Graphics
22. Multiplan v. 1.21
23. Open Access v. 1.0
24. Symphony v. 2.0
25. Framework IIP, kompletny opis + częściowo spolszczony program (R. Wacławek)
26. FRAMEWORK III, polskie litery
27. Eureka
28. Math-Cad
29. First Choice, pakiet zintegrowany, polskie litery
30. Who-What-When v. 1.09, pr. zarządzający

C. EDYTORY TEKSTU, DESKTOP PUBLISHING
1. Chi-Writer v. 2.02
2. Chi-Writer v. 3.11
3. PISMAK, całkowicie spolszczony Chi-Writer v. 2.02, szybkie wydruki (tryb tekstowy)
4. PELIKAN, kompletny opis + całkowicie spolszczony system MS Word v. 3.0 (R. Wacławek)
5. PC Write v. 2.5
6. Wordstar 2000
7. DRUKARNIA, kompletny opis + całkowicie spolszczony system Page Maker, do pracy wymaga OFICYNY (R. Wacławek)
8. DESKTOP PUBLISHING porównanie najbardziej popularnych pakietów

D. SYSTEMY OPERACYJNE, PROGRAMY UŻYTKOWE
1. PC-DOS v. 3.1 i 3.2 (R. Wacławek)
2. DOS 3.3
3. DOS 4.0
4. OS-2
5. OFICYNA, kompletny opis + całkowicie spolszczony system MS Windows (R. Wacławek)
6. Norton Commander 87
7. Norton Commander 89
8. PC Tools De Luxe
9. PC Tools v. 2.02, Lettrix
10. Sidekick P, opis + spolszczony program (R. Wacławek)
11. Drukarz, spolszczony program Lettrix, rozszerzenie możliwości wydruku (R. Wacławek)
12. GOP, programowa instalacja polskich znaków na ekranie i drukarce, współpraca z dowolnym oprogramowaniem

E. CAD / SIECI KOMPUTEROWE
1. Autocad v. 2.17, kompletny opis + częściowo spolszczony program (R. Wacławek)
2. Auto-Cad v. 2.6 i 9.0 (wprowadzenie)
3. Or-Cad v. 1.21
4. KREŚLARZ, spolszczony system projektowania inżynierskiego In-a-vision, praca łącznie z OFICYNĄ (R. Wacławek)
5. Lan-Link v. 4.0 (R. Wacławek)
- 6a. Novell, podręcznik użytkownika
- 6b. Novell, podręcznik instalatora
- 6c. Novell, instalacja karty Arcnet
- 6d. Novell, instalacja karty Ethernet
7. Novell v. 2.12 opis
8. Novell v. 2.15 (dodatek do v. 2.12)
9. D-Link v. 3.24
10. OA-Link
11. WYKRES, opis + spolszczony MS Chart
12. ELEKDRUK, opis + spolszczona wersja programu Smartwork (R. Wacławek)

F. RÓŻNE
1. Przewodnik programisty IBM
2. Wprowadzenie do IBM (R. Wacławek)
3. PC 1512, instrukcja obsługi
4. PC 1512, opis techniczny
5. Locomotive Basic 2 (PC 1512)
6. Poly-Windows
7. Instrukcja drukarki NL-10
8. Instrukcja drukarki SG-15
9. Instrukcja drukarki LC-10
10. Instrukcja drukarki NX-15
11. Instrukcja drukarki NB 24-15
12. PAW, profilaktyka antywirusowa
13. FILEMON, oprogr. myszki (R. Wacławek)
14. Zestaw polskich czcionek do drukarki laserowej.

Zapraszamy do współpracy przy tworzeniu polskiego oprogramowania oraz dokumentacji komputerowej.

Zniżka 10% przy płatności czekiem lub gotówką.

W przygotowaniu:
1. MICROSOFT C v. 5.1
2. QUATRO
3. PAINT BRUSH
4. LOTUS 1-2-3 v. 3.0

Zapraszamy w godzinach od 8 do 16.

Ko-61/05

Kompilatory firmy Jensen & Partners International dostępne już w Polsce

TopSpeed Modula-2 (progr. obiektowe) **TopSpeed C** (100% ANSI, progr. współbieżne)

MERCURIA LTD. - dystrybutor oprogramowania firmy JPI w Polsce - oferuje wyżej wymienione kompilatory w wersji standardowej i profesjonalnej (DOS) oraz profesjonalnej (OS/2).

Wersja **profesjonalna zawiera**: zintegrowany edytor, pomoc w formie HyperTextu, SmartLinker, Visual Interactiv Debugger, debugger "post-mortem"(!), biblioteki z pełnym kodem źródłowym, PROM Locator, TopSpeed: Assembler, Profiler, Disassembler i wiele innych profesjonalnych narzędzi.

Zapewniamy całodobowy serwis techniczny. Otrzymacie Państwo również licencję na sprzedaż programów napisanych przy użyciu zakupionych kompilatorów.

Proszę do nas dzwonić lub pisać w celu otrzymania pełnej specyfikacji przedstawionych pakietów. Oferujemy również dodatkowe biblioteki znanych firm światowych (bazy danych, grafika, sieci, generatory aplikacji itp.)

MERCURIA LTD.
ul. Powstańców Śląskich 95
53-332 Wrocław
telefon: (071) 60.56.79
telefax: (017) 61.67.11
telex: (071) 24.91 opg pl

Co-34/482/06

ASL 2000 PROSTER

ANALIZATOR STANÓW LOGICZNYCH
- 16, 32,48 lub 64 kanały
- sondy typu VAR lub TTL
- pamięć: 2046 stanów/kanał
- zegar wewnętrzny do 80MHz
- wykonanie: terminal PC
  kompakt z PC

PROSTER
Gliwice 44-101, ul. Zygmunta Starego 6
tel. 31-48-60, 31-90-21, 32-06-32
tlx 036315, 036322

ASL 2000

Ko-95/467 06

Wojewódzkie Przedsiębiorstwo Handlu Wewnętrznego Oddział w Tychach

43-100 Tychy, Al. ZMP 77
tel. 27-69-75

VIDEOBIT

poleca między innymi
- **sprzęt komputerowy**
  Atari ● Commodore ● Amstrad ● IBM PC XT/AT/PS-2
- **drukarki STAR, EPSON, AMSTRAD**
- **sprzęt audiowizualny**
  magnetowidy ● OTV PAL/SECAM ● wideoskopy ● kamery
- **anteny satelitarne**
- **aparaturę badawczo-naukową**

Udzielamy gwarancji, prowadzimy naprawy pogwarancyjne.

Zapewniamy o atrakcyjnych cenach.

Ko-76/236/01

UNICOMP ELECTRONICS EXPORT

Salon Sprzedaży EUROPA

Centrum Warszawy – ul. Armii Ludowej 15 – telefon: 25-76-94

Sprzedaż za złotówki. Polecamy:

1) Komputery domowe

ATARI 520 STFM		5.995.000,-
AMIGA 500 + modulator i mysz		6.995.000,-
SAM COUPE (280) + stacja dysków		4.155.000.-

2) Drukarki

D-100 Mera-Błonie	25cm/100zn/s	1.495.000,-
MENNESMANN TALLY MT-81	25cm/160zn/s	2.945.000,-
PANASONIC KX-P1180	30cm/200zn/s	3.695.000,-
PANASONIC KX-P1124 24igły	30cm/200zn/s	6.335.000,-
PANASONIC KX-P1592	40cm/220zn/s	6.335.000,-
PANASONIC KX-P1595	40cm/300zn/s	inf. telefon
PANASONIC KX-P1540 24igły	40cm/240zn/s	inf. telefon
FUJITSU DX 2300	25cm/320zn/s	10.635.000,-
FUJITSU DX 2400	40cm/320zn/s	12.995.000,-
FUJITSU DX 3400 24igły	40cm/240zn/s	14.615.000,-

Drukarki laserowe

FUJITSU RX 7100 PS 5 str/min	inf. telefon
MANNESMANN Tally Laser	inf. telefon
STAR LASER 8 8 str/min	inf. telefon
ATARI LASER SLM 804	inf. telefon

3) Stacje dysków

do AMIGI 5,25 cala	2.695.000,-
do AMIGI 3,5 cala	1.595.000,-
do ATARI 5,25 cala	2.285.000,-
do ATARI 3,5 cala	1.695.000,-

4) Dyskietki

dyskietki firmowe:	5,25 DSDD	10 szt	145.000,-
	5,25 DSHD	10 szt	245.000,-
	3,5 DSDD	10 szt	245.000,-
	3,5 DSDD	1 szt	15.000,-

5) Komputery przenośne

HITACHI HL 320-021 – typu Laptop 80C88 – 4.77 MHz, RAM 512 kB, LCD 640x200, FDD 1x720, HDD 20 MB, Centronics, RS-232, MS-DOS 3.2 — inf. telefon

ATARI PORTFOLIO – podręczny komputer typu IBM XT 80C88 – 5 MHz, RAM 128 kB, ROM 256 kB LCD 240x64, z wbudowanym programem typu MS-DOS, LOTUS, baza danych, inne, z możliwością rozbudowy RAM do 640 kB, RS, Centronics — 5.295.000,-

6) Komputery profesjonalne:

na zamówienie płatne w dolarach (przelew) lub w złotówkach — inf. telefon

*** Komputery typu IBM PC/XT/AT/386**

VIP XT 8088-10 MHz, RAM 640 kB, karta MGP, FDD 2x360, zasilacz 150W, zegar, Centronics, RS-232, klawiatura 101

VIP AT 80286 – 12 MHz, OWS, RAM 1 MB (możliwe 4 MB na płycie głównej Suntac), ems, karta MGP, FDD 1,2 i 360, zasilacz 180W, zegar, Centronics, RS-232, klawiatura 101

VIP 386 AT 80386 – 20 MHz, OWS, RAM 2 MB, karta MGP, FDD 1,2 i 360, zegar, Centronics, RS-232, zasilacz 200W, klawiatura 101

⋆ **Rozbudowa pamięci RAM**
w XT do 1 MB ● w AT do 2 MB ● w AT do 4 MB

⋆ **Dyski stałe** Tylko w zestawie z komputerem
20 MB/60ms ST-225 ● 40 MB/28ms ST-251 ● 80 MB/28ms ST-4096

7) Karty do komputerów (na zamówienie czas realizacji 4 tygodnie)

Karta prototypowa PC1 (bufor danych, bufor i dekoder adresów)	250.000,-
Karta prototypowa PC/AT1 (bufor danych)	350.000,-
Karta we/wy-1/O 8255PC (2 szt 8255, 1 szt 8253)	450.000,-
Karta 1xRS-232-RS2 (jako RS 1, 2, 3, 4)	250.000,-
Karta 2xRS-232-RS2 (jako RS 1, 2, 3, 4)	350.000,-
Karta 4xRS-232-RS4 (jako RS 1-4, 5-8)	900.000,-
Karta 8xRS-232-RS8	1.400.000,-
Karta programatora EPROM PCP-512 2716-27512 wraz z polskim programem	800.000,-
Karta drukarki Centronics-PPC (jako LPT 1, 2, 3)	250.000,-
Karta sumulatora EPROM 2, 4, 8 kB SYM-8	350.000,-
Karta przetwornika AD/DA 13 bit, 30 ms, 16 wejść/8bit, 30us, 4 wejscia	900.000,-
Karta IEEE-488	900.000,-

Wszystkie artykuły z roczną gwarancją. Serwis: BŁONIE, ul. Przybysza 20 telefon: 554-554

Co-30/475/06

eurobit

01-571 Warszawa
ul. Kozietulskiego 8/12
tel. 39 66 02, 39 66 17
tlx 816255 ebit pl

OFERUJE!

DOSTAWY NIEZAWODNEGO SPRZĘTU

mikrokomputery: IBM PC/XT/AT/386

przykładowe ceny:

1. XT (Hercules, monitor 14") – 5300000,-
2. XT (j.w. + HD 20 MB) – 8200000,-
3. AT (Hercules, monitor 14") – 7200000,-
4. AT (j.w. + HD 40 MB) – 10400000,-
5. 386 MONO HD 80 MB – 21000000,-

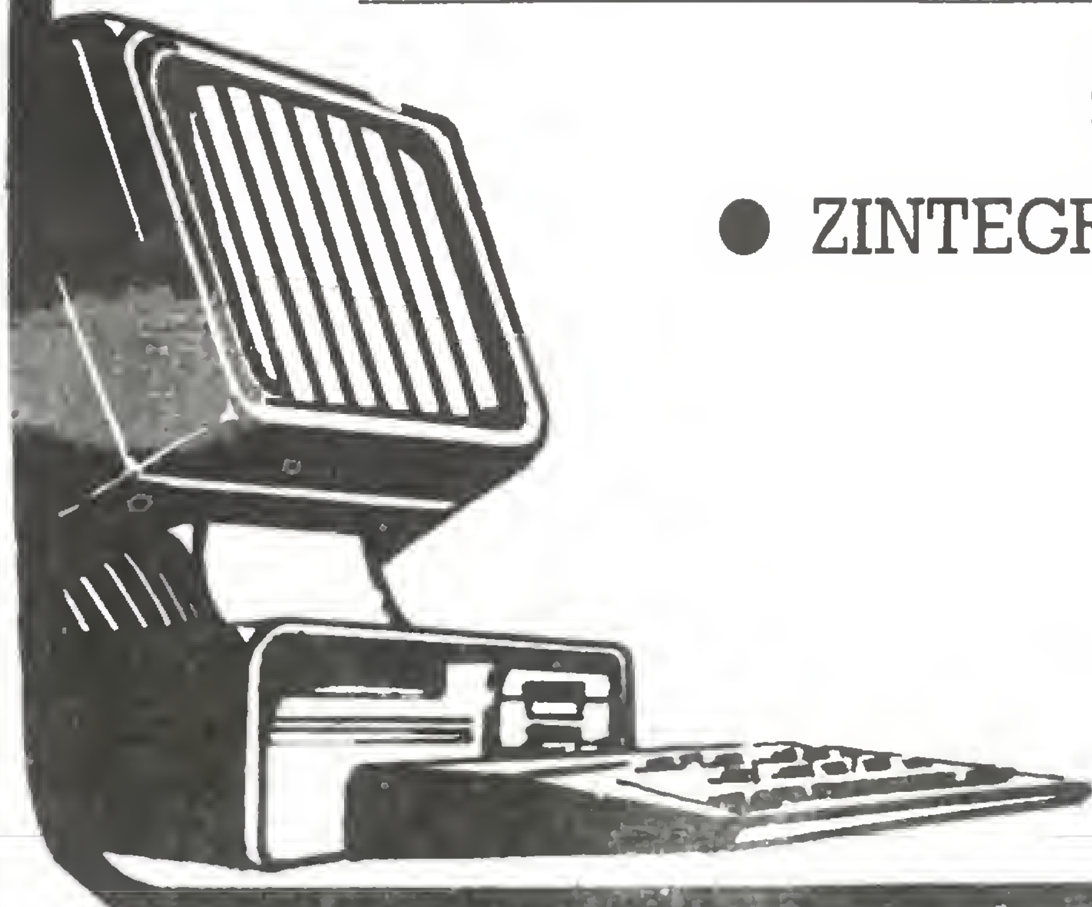
drukarki:

1. STAR LC 15 (180 zn/s) – 4800000,-
2. STAR FR 15 (300 zn/s) – 7400000,-

oraz materiały eksploatacyjne, obsługa gwarancyjna i pogwarancyjna (również sprzętu zakupionego w innych firmach)

POLSKĄ DOKUMENTACJĄ DO PC XT/AT

- dBASE IV – pełny opis
- TURBO PASCAL v. 5.0
- CLIPPER '87
- DOS 3.30; DOS 4.0
- Drukarki Star: NX15, LC10, LC15, FR 10/15
- i inne – na życzenie przesyłamy katalog



SYSTEMY KOMPUTEROWE NA PC XT/AT

● ZINTEGROWANY PAKIET ZARZĄDZANIA PRZEDSIĘBIORSTWEM ●
składający się z następujących modułów:

- system "Obsługa Biura Handlu Zagranicznego"
- system finansowo-księgowy
- system gospodarki materiałowej
- system płacowy i kadrowy
- system obsługi transportu

Również instalacja na ICL DRS – 300

Co-27/457/06

W październiku miałem okazję z bliska zobaczyć francuski rynek komputerowy.

Ogólnie można stwierdzić, że nasycenie sprzętem komputerowym w biurach, urzędach, placówkach handlowych, hotelach, bankach, biurach podróży jest znacznie większe niż w naszym kraju. Właściwie to u nas komputerów w takich miejscach nie widać, we Francji widać je wszędzie. W większości są to maszyny zgodne ze standardem IBM PC. W większych biurach komputery połączone są ze sobą siecią lokalną. Często "klony" IBM stanowią inteligentne terminale współpracujące z komputerem większym. Tak jest gdy dane biuro jest modernizowane lub rozbudowywane. Gdy instaluje się sieć komputerową od początku to powszechnie jednostkę centralną i zespół pamięci zewnętrznej stanowi nowej generacji minikomputer pracujący pod kontrolą systemu Unix. Do takiego komputera dołącza się terminale nierzadko dysponujące szerokimi własnymi możliwościami przetwarzania danych.

Zakup komputera we Francji jest sparwą łatwą. Komputery domowe, a właściwie do własnego domowego użytku kupuje się najczęściej w stoiskach z elektroniką domów towarowych, lub pewniej w salonach ze sprzętem audio-wideo. W każdym takim salonie wydzielone jest stoisko ze sprzętem i oprogramowaniem komputerów. Maszyny w typowych zestawach nabywa się "od ręki" a te z nietypowym zestawem elementów można zamawiać u sprzedawcy i wtedy oczekuje się na nie około miesiąca. Największą ofertę stanowią komputery domowe jak Amstard 6128, cała rodzina Atari ST, Commodore C64 i najprostsze wersje Amstradów PC 1640. Do tego sprzętu oferowane są różne urządzenia peryferyjne jak modemy, drukarki, monitory oraz różnorodne akcesoria. Oferta oprogramowania to głównie gry dla komputerów domowych. Dla klasy PC można kupić najpopularniejsze programy użytkowe jak edytory tekstu, kompilatory języków programowania, bazy danych, elektroniczne arkusze obliczeniowe, a także programy edukacyjne. Programy edukacyjne dla najmłodszych dostępne są głównie dla komputerów domowych..

Zakupy sprzętu do wyposażenia biur czy przedsiębiorstw dokonuje się w wyspecjalizowanych firmach. Podobnie jak u nas, z tą różnicą, że sprzedawca zawsze oferuje pomoc w instalacji sprzętu i dobraniu niezbędnego oprogramowania. Szeroko rozwinięta jest działalność firm proponujących kontrakt "pod klucz". Firma taka przed załatwieniem transakcji poznaje specyfikę pracy danego przedsiębiorsta, a następnie proponuje rozwiązania programowo-techniczne. Po uzgodnieniu szczegółów frma zajmuje się kompletacją sprzętu i oprogramowania, instalacją i uruchomieniem systemu. Prowadzi także szkolenie pracowników. Usługi kompleksowe są bardzo kosztowne, ale powszechnie wykorzystywane.

W stoiskach ze sprzętem komputerowym widziałem duże zainteresowanie komputerami przenośnymi typu laptop, faxami oraz najnowszym modelem komputera IBM PS/1.

Po tych kilku uwagach dalej jak zwykle w tym miejscu cennik.

Atari 520 STFM z monitorem mono	5390 FF
Atari Portfolio	1590 FF
Atari VCS 2600 (zestaw gier telewizyjnych) z joystickiem i 30 grami	380 FF
Amstard 6128 z kolorowym monitorem	2690 FF
Commodore Amiga 500 bez monitora	3290 FF
Commodore PC 30 z dyskiem twardym 20 MB i monitorem kolorowym EGA	11900 FF
Amstard PC 1512 z jedną stacją dyskietek i mono monitorem	4265 FF
Amstrad PC 1640 z dyskiem twardym 30 MB i monitorem kolorowym EGA	8990 FF
IBM PS/1 z dyskiem twardym 30 MB i kolorowym monitorem	16970
IBM PS/1 z mono monitorem i jedną stacją dyskietek 3,5 calowych	8990 FF
Bondwell B200 (laptop) bez dysku twardego, ekran ciekłokrystaliczny CGA	4990 FF
Drukarka Citizen 180 (9 igieł)	1690 FF
Drukarka Amstrad DMP 2160	1390 FF
Drukarka Epson LQ 400 (9 igieł)	2590 FF
Tandy 1400 FD (laptop) 2 stacje dyskietek 3,5 cala, ekran ciekłokrystaliczny CGA	11750 FF
Tandy 1000 SL/2 z monitorem kolorowym i kartą CGA	9485 FF
Monitor Tandy VM-22 mono bursztynowy	965 FF
Monitor Tandy VGM-300 kolorowy VGA, rozdzielczość punktu świecącego 0,3 mm	5337 FF
Dysk twardy Tandon 20 MB z kontrolerem	3290 FF
Coprocesor 80287 - 16 MHz	3480 FF
Joysticki	33 - 129 FF
Pudełko na 100 sztuk dyskietek 5,25 cala	69 FF
Dyskietki 3,5 cala MF2DD 10 szt.	89 FF
Dyskietki 5,25 cala 2DD 10 szt.	39 FF

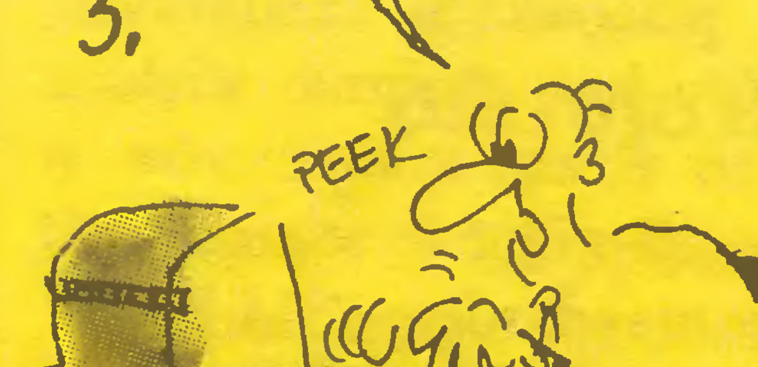
Dla urealnienia cen podaję, że jednorazowy bilet do metra kosztuje 2 FF, zapiekanka w barze na ulicy ok. 14 FF, litrowa butelka wina stołowego ok. 29 FF, obiad firmowy w typowej restauracji ok. 90 FF, jedna noc w hotelu klasy Holiday Inn 1500 FF, minimalna płaca 5600 FF miesięcznie. Dodatkowo podaję, że we Francji rozwinięta jest bardzo sprzedaż ratalna. Minimalna rata za sprzęt komputerowy wynosi 300 FF. W czasie gdy zbierałem te dane za 1 dolara USA płacono ok. 4,87 FF, a za 1 FF należało zapłacić 1830 zł.

Zebrał Z.R.

rys. P. Kakiet



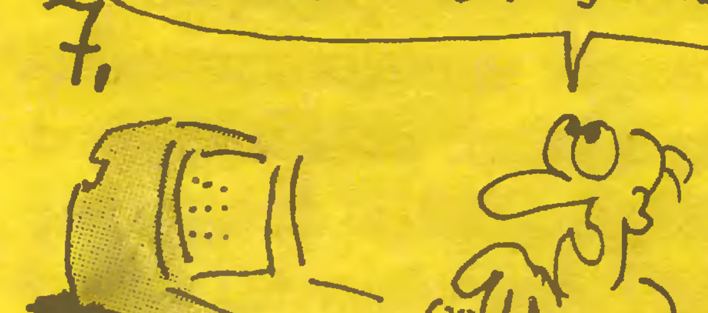


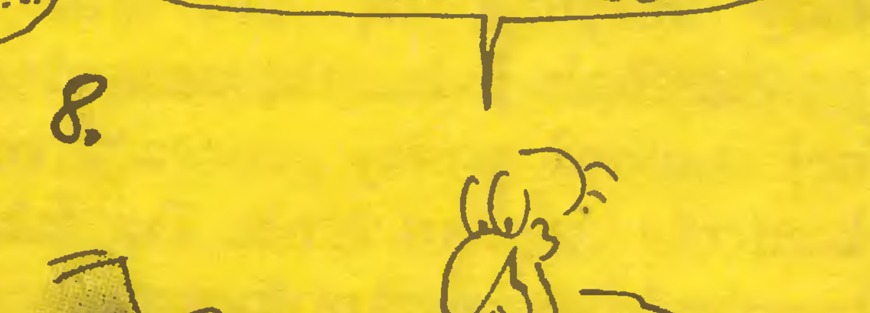


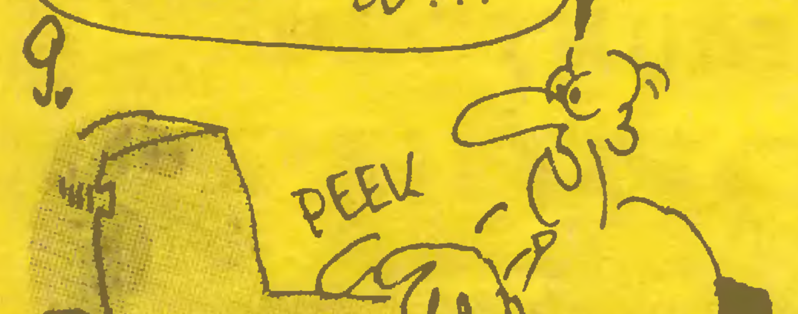










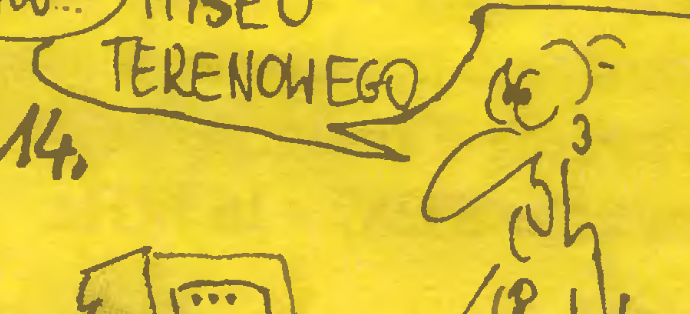


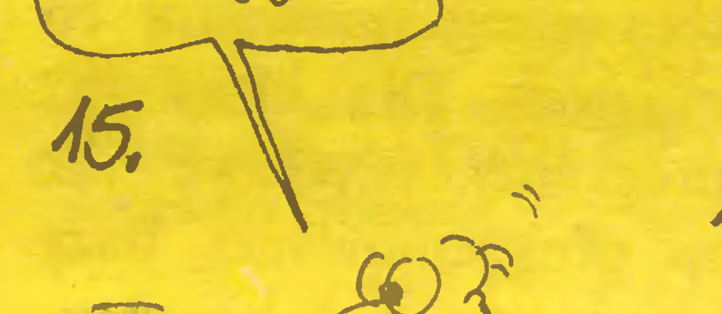


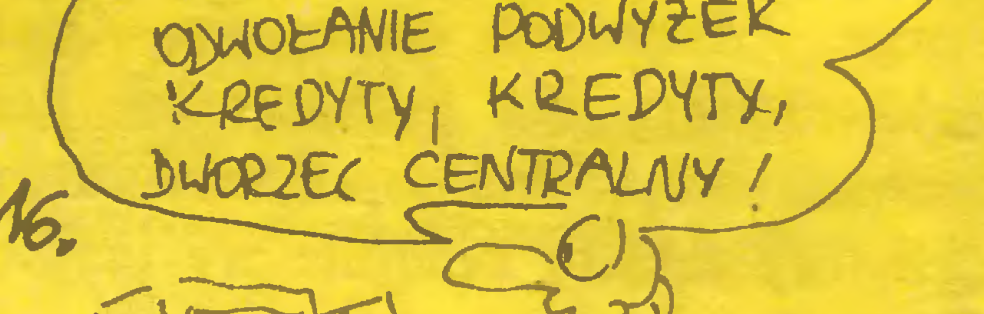




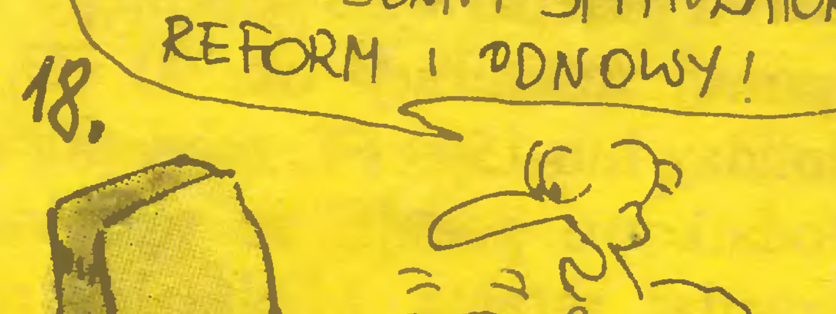


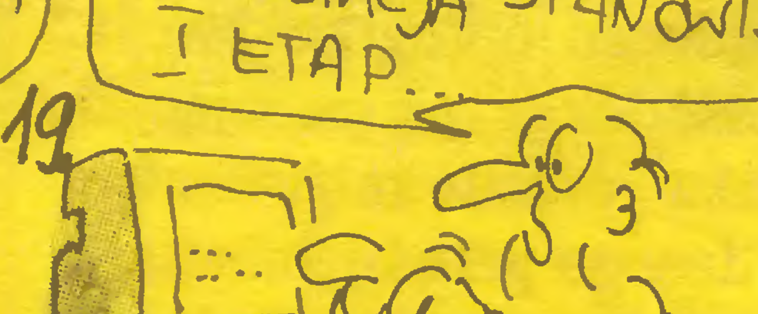


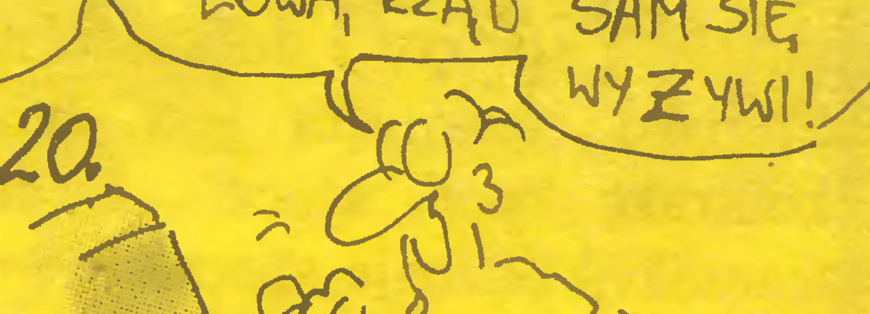


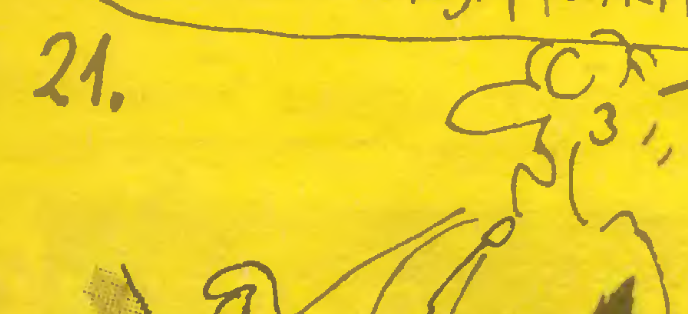


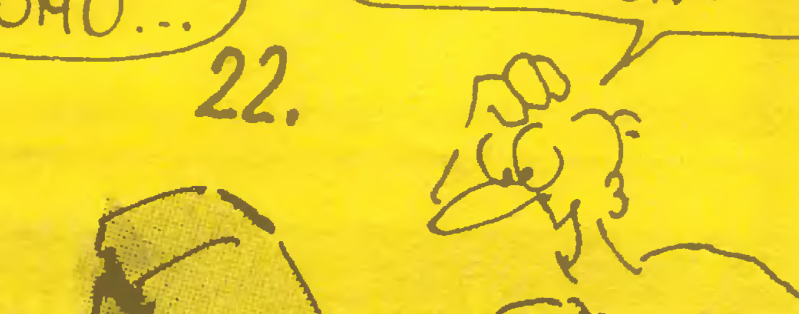


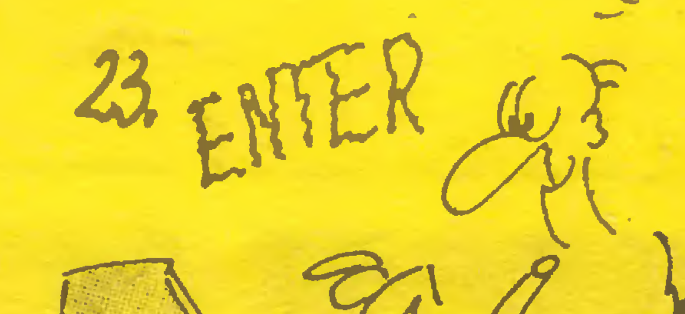


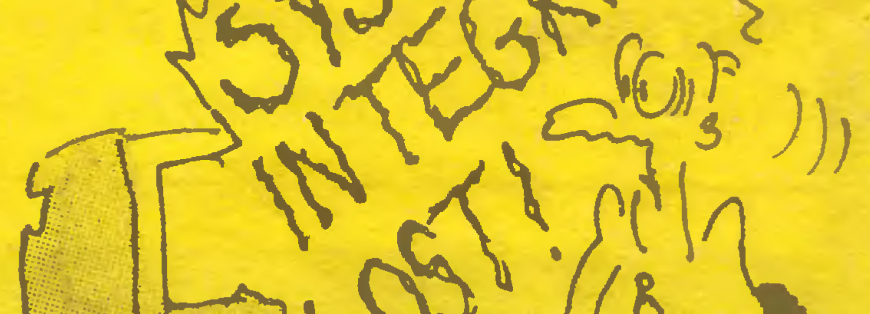












TEST KOMPUTERA IBM